



Diploma Thesis

# Numerical Studies on Crack Arrays in Aluminum Pressure Die Casting Molds

by  
Peter Raninger

Montanuniversität Leoben

&

Materials Center Leoben



Leoben, November 16, 2009

### **Affidavit**

I hereby declare in lieu of oath that the following diploma thesis "Numerical Studies on Crack Arrays in Aluminum Pressure Die Casting Molds" has been written only by the undersigned and without any assistance from third parties. Furthermore, I confirm that no sources have been used in the preparation of this thesis other than those indicated in the thesis itself.

---

Place, Date

---

Signature

I am especially grateful for their support and helpful suggestions to

- The supervisors of my thesis Prof. Dr. Thomas Antretter and Prof. Dr. Reinhold Ebner.
- My advisor Dr. Werner Ecker who introduced me to the current topic and always gave me support in the "dark hours" of crack array simulation.
- Dr. Mario Leindl whose doctoral thesis and advice were fundamental for this work.
- Prof. Dr. Reinhard Pippan and Prof. Dr. Otmar Kolednik for their valuable insights in their areas of expertise.
- All my dear colleagues whose advise and friendship are very precious to me.
- Last but not least I want to express my gratitude towards the Materials Center Leoben Forschung GmbH where this thesis was carried out within the scope of an industrial project.

## Abstract

Pressure casting dies are exposed to harsh service conditions consisting of thermal and mechanical loading and thus undergo thermo-mechanical fatigue. Due to cyclic plastic deformation of the material near the surface of the dies the loading conditions gradually change because of the formation of tensile residual stresses which add to the stress field from external loading. This change in the stress field influences the nucleation and growth of cracks. Typically after a few thousand casting cycles a network of heat checks forms. Within a crack network crack shielding has a big influence on the evolution of the crack array. The crack spacing within the network of heat checks, the length at which the cracks stop growing and the crack growth rate are influenced by the thermo-physical and mechanical properties of the die material. In the present study the influences of the material properties on the formation and growth of heat check networks are investigated in a numerical study on the evolutions of crack arrays.

In order to obtain the loading conditions for the numerical study on crack arrays a finite element (FE) model of a flat surface area of the pressure casting die is generated to calculate the transient temperature field and the evolution of residual stresses over 100 casting cycles. The temperature field and the residual stress field of cycle 100 are used in script based FE-models with linear elastic material behavior investigating the formation and evolution of crack arrays. By the aid of a Python script FE-models are created, computed and evaluated automatically. In this way a modeling chain is formed in which data obtained from the previous model are used to create the subsequent one. For this purpose, the script compares the maximal stress intensity factor  $K_{max}$  to a user defined criterion. In this criterion  $K_{max}$  is converted, based on experimental data and parallel computation of stress ratios, to the effective stress intensity factor  $\Delta K_{eff}$  and compared to a threshold value  $\Delta K_{th}$ , which is a measure for crack arrest. In this way an initial crack configuration and the evolution of crack arrays can be modeled in an automated manner for different materials.

The focus in this work is put on the interplay of material parameters and crack shielding and on the interconnection between hardness, thermal conductance and thermal expansion. The ultimate goal in this work is to provide mathematical relations linking material parameters and crack propagation that can be used for material selection and material development.

## Kurzfassung

Die im Aluminium Druckguss eingesetzten Gussformen sind einer Ermüdungsbeanspruchung infolge thermischer und mechanischer Belastungen ausgesetzt. Zyklisch plastische Verformung an der Oberfläche der Formen bewirken eine schrittweise Veränderung des Belastungsprofils, da sich Zugeigenspannungen aufbauen, die sich den Spannungen zufolge der zyklischen Belastungen überlagern. Diese Veränderung des Spannungsfeldes hat Einfluss auf Bildung und Wachstum von Rissen. Nach einigen Tausend Lastzyklen beginnt sich typischerweise ein Netzwerk aus Heißrissen auszubilden, in welchem die gegenseitige Abschirmung der Risse wesentliche Auswirkungen auf die weitere Entwicklung des Netzwerkes hat. Große Bedeutung wird auch den thermo-physikalischen und den mechanischen Eigenschaften des verwendeten Materials für die Gussformen beigemessen, da diese die Anfangsabstände, die Tiefe, in der die Risse stoppen, und die Risswachstumsgeschwindigkeiten der Risse beeinflussen. In dieser Arbeit werden die Einflüsse der Materialparameter auf die genannten Merkmale und Ausbildung der Heißrissnetzwerke numerisch untersucht.

Um die Belastung der Gussformen, die für die Simulation der Rissnetzwerke benötigt werden, zu bestimmen, wird ein Finite Elemente (FE) Modell eines ebenen Abschnittes der Form erzeugt, um das stationäre Temperaturfeld sowie den Aufbau der Eigenspannungen über 100 Gießzyklen hinweg zu berechnen. Die so erhaltenen Verteilungen für Temperatur und Eigenspannungen werden an ein script-basiertes FE Modell übergeben, welches erlaubt, unter Annahme von linear-elastischem Materialverhalten, die Entstehung und Ausbildung von Rissnetzwerken zu untersuchen. Unter Zuhilfenahme eines Python-Scripts werden einzelne FE Modelle automatisch erzeugt, berechnet und ausgewertet. Dadurch entsteht eine Kette, in der Daten eines vorhergehenden Modells verwendet werden, um das darauf folgende zu generieren. Zu diesem Zweck überprüft das Script ein implementiertes Kriterium. Bei diesem wird  $K_{max}$ , auf Basis experimenteller Daten und parallel gerechneten Modellen, in eine effektive zyklische Spannungsintensität  $\Delta K_{eff}$  umgerechnet und mit einem Schwellwert  $\Delta K_{th}$  verglichen. Somit können die Bereiche, in denen Risse stoppen, identifiziert werden. Mit dieser Vorgehensweise können Anfangskonfigurationen und die Ausbildung der Netzwerke für verschiedene Materialien automatisiert durchgeführt werden.

Die Schwerpunkte dieser Arbeit liegen auf der gegenseitigen Beeinflussung von Materialparametern und Rissabschirmung, sowie die Wechselwirkungen untereinander zwischen Härte, Wärmeleitfähigkeit und Wärmeausdehnung. Letztendlich sollen mathematische Zusammenhänge zwischen Materialparametern und Risswachstum gefunden und der Werkstoffwahl und Werkstoffentwicklung zur Verfügung gestellt werden.

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
1.1	Overview . . . . .	4
1.2	State of the Art . . . . .	5
1.2.1	Physical Simulation of Heat Check Formation . . . . .	5
1.2.2	Material Modeling . . . . .	6
1.2.3	Analysis of Loading Behavior . . . . .	7
1.2.4	Simulation of Crack Arrays . . . . .	8
1.3	Current Problem Definition . . . . .	10
<b>2</b>	<b>Fundamentals of Fracture Mechanics</b>	<b>11</b>
2.1	Linear Elastic Fracture Mechanics . . . . .	12
2.1.1	Energy Approach . . . . .	12
2.1.2	Stress Intensity Approach . . . . .	16
2.2	Elastic Plastic Fracture Mechanics . . . . .	19
2.2.1	CTOD . . . . .	19
2.2.2	J-Integral . . . . .	20
2.3	Fatigue Basics . . . . .	23
2.3.1	Damage-Tolerant Approach . . . . .	24
2.3.2	Total Life Approaches . . . . .	26
2.4	Near Threshold Fatigue . . . . .	28
2.5	The Fatigue Threshold . . . . .	30
2.6	Small Fatigue Cracks . . . . .	32
2.7	Crack Closure . . . . .	36
2.7.1	Overview . . . . .	36
2.7.2	Plasticity Induced Crack Closure . . . . .	38

<i>CONTENTS</i>	2
2.8 Crack Arrays . . . . .	40
2.8.1 Fundamentals . . . . .	40
2.8.2 Former Studies on Crack Arrays . . . . .	41
<b>3 Modeling</b>	<b>48</b>
3.1 Preliminary Studies: T-Model and S-Model . . . . .	48
3.1.1 Model Design . . . . .	48
3.1.2 Generation of Input Files . . . . .	50
3.1.3 Temperature Distributions . . . . .	51
3.1.4 Residual Stresses . . . . .	51
3.1.5 Data Processing I . . . . .	52
3.2 The Architecture of Abaqus . . . . .	53
3.3 Main Studies: Control Script and C-Model . . . . .	57
3.3.1 Determination of the Stress Ratio . . . . .	59
3.3.2 Determination of the Effective Stress Intensity Range . . . . .	61
3.3.3 The Relation between Crack Propagation and Load Cycles . . . . .	65
3.3.4 Parametric Studies . . . . .	68
3.3.5 Determination of the Characteristic Minimum Spacing . . . . .	68
3.3.6 Data Processing II . . . . .	69
<b>4 Results</b>	<b>71</b>
4.1 Temperature Distribution . . . . .	71
4.2 Formation of Residual Stresses . . . . .	73
4.3 Stress Ratio . . . . .	77
4.4 Crack Shielding . . . . .	79
4.5 Evolution of Crack Arrays for Standard Parameters . . . . .	81
4.6 Study on Service Life Time with Constant Initial Spacing . . . . .	89
4.7 Influence of Material Parameters on Crack Growth with Variable Spacing . . . . .	95
4.7.1 Possible Influences on the Initial Crack Spacing . . . . .	95
4.7.2 Study on Service Life Time with Variable Initial Spacing . . . . .	96
4.8 Limitations of the C-Model . . . . .	108
<b>5 Discussions and Conclusions</b>	<b>109</b>

<i>CONTENTS</i>	3
<b>6 Appendix</b>	<b>124</b>



# Chapter 1

## Introduction

### 1.1 Overview

Aluminum pressure die casting is one of the most economic ways to process aluminum. The dies are exposed to harsh service conditions and thus prone to heat checks, erosion and other damage. New concepts are supposed to increase economic gain by extending the service life time of the dies<sup>1</sup>.

According to [1] the predominant causes for degradation of the dies are:

- Thermal fatigue due to the interplay of heating and cooling in each casting cycle.
- Corrosion due to oxygen, metals, mold release agents, etc.
- Erosion due to focused beams of melt and particles transported with them.
- Softening due to tempering under service conditions.
- The technique used for finishing, like spark erosion, machining and polishing.

The formation of single cracks can cause early failure since propagation rates are very high and the cracks can reach a critical size in a few thousand cycles. Furthermore it was found that service life is strongly dependent on a thin surface layer of usually 2 mm or less in case of the formation of crack networks. Thermal fatigue is of essential meaning for the formation of crack arrays and single crack growth and thus it can be regarded as

---

<sup>1</sup>The term die is equivalent to mold but used specifically for casting, whereas molds are also used e.g. in forging.

the main factor limiting the service life of casting dies.

The mechanical properties of the die material and their temperature dependence are essential in view of thermal fatigue. Temperature gradients due to cyclic heat flow cause cyclic stresses which induce a plastic material response in the surface region. Due to the plastic cyclic strain the loading conditions gradually change because of the formation of tensile residual stresses which add to the stresses from thermal loading. Thus, the mean stress is shifted to larger values and damage is accumulated during each casting cycle causing conditions favoring the nucleation and growth of cracks to form at the surface [2]. The typical materials used in aluminum pressure casting are hot work tool steels like 1.2343 (X38CrMoV5-1) or 1.2367 (X38CrMoV5-3), where the latter is resistant to higher thermal loads [2]. All those grades are refined by Electro-Slag-Remelting or Vacuum-Arc-Remelting, which give the material enhanced mechanical properties. Surface treatments like nitriding and coating are applied but those are restricted to smaller die geometries. For more demanding applications alternative types like maraging steels and molybdenum alloys can be used [3].

The material investigated in this thesis and in the work preceding this thesis is the hot work tool steel grade 1.2343 (X38CrMoV5-1) manufactured by Böhler Edelstahl GmbH & Co KG (internal steel grade W300).

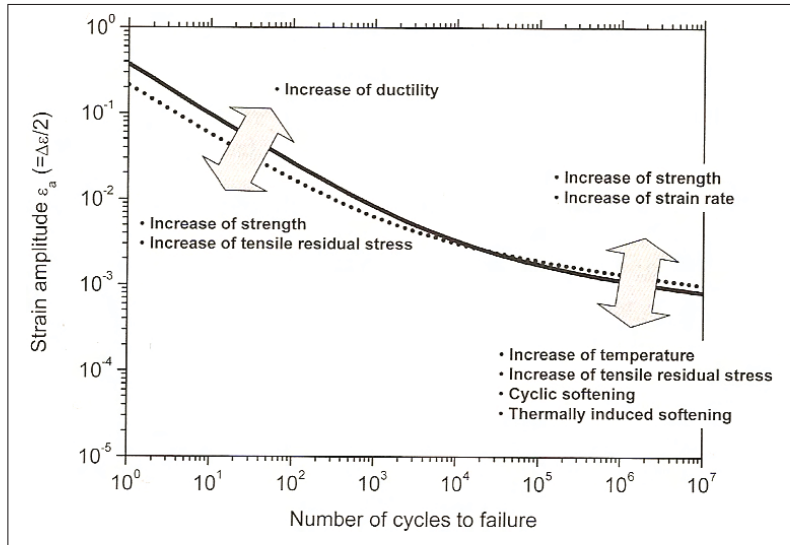
## 1.2 State of the Art

### 1.2.1 Physical Simulation of Heat Check Formation

At the Materials Center Leoben a testing facility was designed to study the formation of crack arrays under conditions similar to service conditions of aluminum pressure casting dies. This facility utilizes pulsed laser radiation in order to subject samples to cyclic thermal loads. The laser provides a high grade of flexibility which is convenient to approximate the temperature evolution occurring in the dies. The facility was used to compare thermal fatigue behavior of different steel grades and different heat treatments. In all samples investigated rough surfaces were found due to plastic deformation and the formation of microcracks.

In order to find the appropriate testing conditions, numerical simulations of the casting process were done. This way, the surface temperatures in the solidification step were determined and used for the settings of the laser. The finite element (FE) simulations also showed that an increase in thermal conductivity causes lower maximum temperatures and smaller temperature gradients. By the laser pulse experiments it was shown that higher

hardness increases lifetime in case of small strain ranges but has a negative effect, or none at all, in case of higher strain ranges (Figure 1.1). In addition, it was found that crack nucleation starts at the grain boundaries and that thermal fatigue behavior of martensitic microstructures are superior to mixed martensitic/bainitic microstructures. Further details can be found in [2].



**Figure 1.1:** Strain based Wöhler curve illustrating the correlation of service life and hardness [2]. A larger hardness is favorable in case of small strains but has a negative effect or none at all if larger strain ranges prevail.

## 1.2.2 Material Modeling

Beside experimental testing, an advanced elastic viscoplastic Chaboche-type model describing the material response to thermal loading was used by Ecker and calibrated with test data on 1.2343 (X38CrMoV5-1) steel grade [4]. The material model was implemented in a FE model allowing to take into account various material phenomena like the Bauschinger effect, cyclic hardening/softening with elastic or plastic shakedown, ratcheting, relaxation of mean stress, rate dependence, creep and relaxation. The model was calibrated with data from isothermal compression tests<sup>2</sup> and verified by the laser pulse experiments. Residual stresses calculated in FE models are matching the experimental

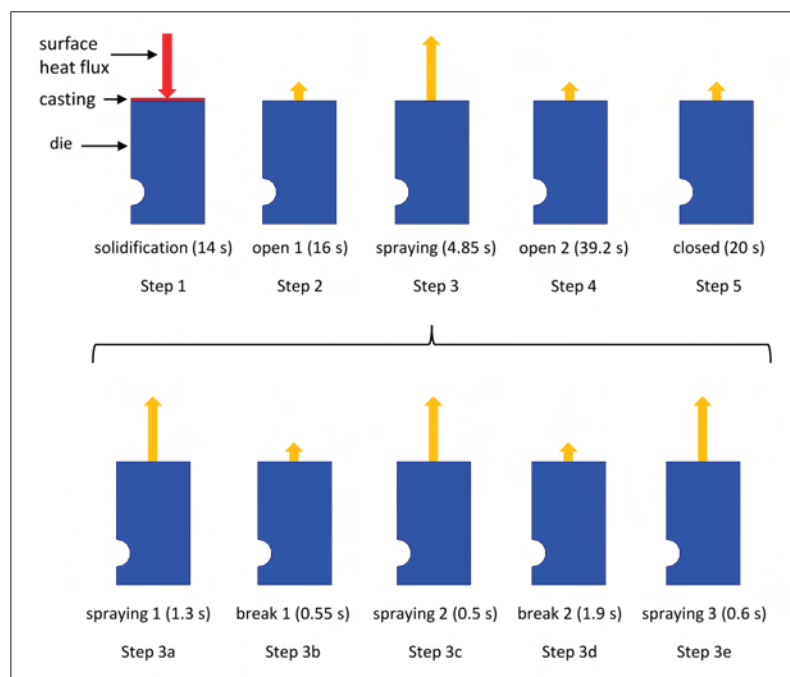
<sup>2</sup>Providing data on isothermal compression tests is required to describe the isotropic and kinematic hardening/softening of the material

results well. Thus, the material model can be utilized for the prediction of stresses and strains in the dies [5, 6].

### 1.2.3 Analysis of Loading Behavior

In order to make use of the material model an analysis of a casting cycle was carried out. A FE model was designed consisting of two die halves and the melt. The next step was to model the process itself and thus a complete casting cycle was divided into five distinct steps and five substeps, where specific parameters were set in each of them. The steps and their corresponding heat flux vectors are schematically shown in Figure 1.2. With the FE model it is possible to compute temperature distributions in the dies for an arbitrary number of casting cycles, and by means of the material model the corresponding evolution of residual stresses can be determined.

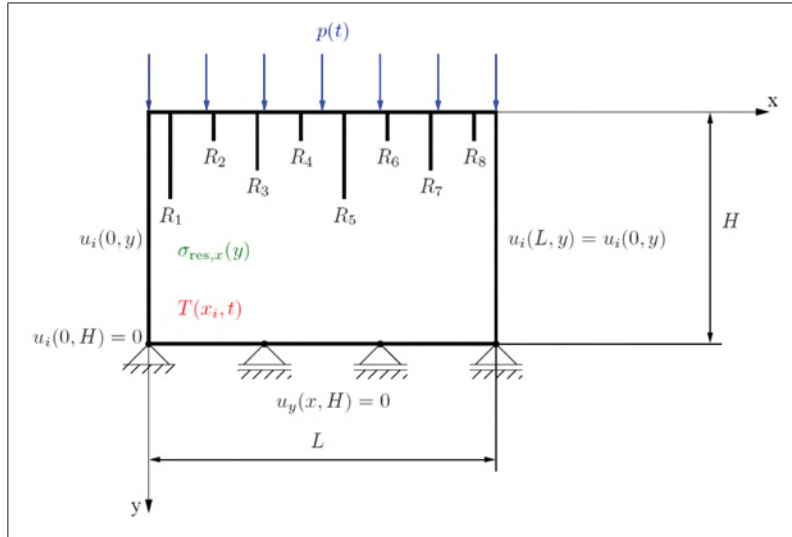
So far, the loading and the material response has been characterized, but crack growth was not an issue at this point. The work of Leindl [7] deals with this part of the project.



**Figure 1.2:** Steps of a pressure die casting cycle [4]. The heat flux vectors are indicated by arrows.

## 1.2.4 Simulation of Crack Arrays

In a next step an FE model was designed and utilized for the simulation of crack growth by Leindl [7]. The FE analysis was split into two parts, one for the determination of temperature and stress distributions using the material laws discussed in the previous section and one for the analysis of crack propagation itself. These models were used to compute 17 different configurations of the crack array, starting with cracks of equal length (0.1 mm) and ending with a specific final configuration with one crack reaching a length of 4 mm. The initial crack distance was determined from crack patterns observed on ingots provided by the industrial partner. The crack spacing was found to be close to 2 mm. Figure 1.3 shows a typical crack configuration and illustrates the periodic boundary conditions.



**Figure 1.3:** Sketch of a typical crack configuration and the boundary conditions used in the FE analysis [7].

Crack growth was described by an extended Paris equation [8] using experimentally determined parameters. In the model crack propagation was characterized by integrating this extended Paris equation based on data obtained from the 17 crack configurations. In the course of the simulations the stress ratios and  $\Delta K_{eff}$  were computed. Thus, the number of casting cycles for reaching the final configuration could be determined.

The 17 configurations of the crack array were computed for several materials whose material parameters are fictional but based on a standard material. The thermal conductivity ( $\lambda$ ) and the heat expansion ( $\alpha$ ) were varied based on the material parameters of the X38CrMoV5-1. This way it was possible to study the impact of  $\lambda$  and  $\alpha$  on service life,

which led to knowledge essential to casting industry and further research.

It was found that the consideration of the residual stresses and thus the material model discussed above is absolutely necessary to obtain precise results. The simulations for standard material parameters resulted in a service life time very close to values reported from casting industry. Furthermore, it was shown that increasing values of  $\lambda$  and decreasing values of  $\alpha$  prolong service life, where the impact of  $\alpha$  is by far larger. The term service life in this context is related to the number of casting cycles needed for a crack to reach a length of 1.5 mm. This is a typical crack length observed in dies at the end of use provided from industry in combination with data on casting cycles. Investigating the number of service cycles needed to reach a certain crack penetration depth observed in real dies, allows to study the impact of various material parameters on the overall service life time. The number of casting cycles a mold can be used is determined by other factors as well but still it can be related to crack penetration depth.

The results obtained from simulations of crack arrays have revealed the influence of  $\lambda$ ,  $\alpha$ , crack shielding and the parameters  $C$  and  $m$  in the Paris equation. However, it is recommended to always consider the full bandwidth of consequences ensued by modifying a single parameter [9]. An increase in  $\lambda$ , for instance, may be achieved by a reduction of certain solute elements in the hot work tool steel. These elements are essential, however, in view of ductility, yield strength and fracture toughness. Thus, it is crucial to keep an eye on the whole picture. Further aspects on this topic are discussed in [9] and [10].

Although the work of Leindl [10] covers a wide range of investigations there are still questions unanswered. The study on crack propagation was done for a constant crack spacing only. It is still unknown whether the initial crack spacing is dependent on loading conditions and thus material parameters like  $\lambda$  and  $\alpha$ . The dependence of service life on the initial configuration of the network is another issue not yet analyzed. Crack growth itself was characterized for a limited range of materials only ( $\lambda \pm 15\%$ ) and the interplay of  $\lambda$  and  $\alpha$  has not been investigated. Another point not included in [10] is the evolution of crack arrays on the basis of criteria derived from fracture mechanics. The depths where cracks stop were specified in advance based on observations of real crack networks. The evolution of the arrays, however, is dependent on the initial spacing and loading conditions. Thus, an automated and criterion-based simulation of crack arrays is necessary for studies on a larger scale.

### 1.3 Current Problem Definition

The focus in this work is put on the interplay of material parameters and crack shielding. The conclusions drawn from results discussed above do not consider the full influence of crack shielding. It was assumed that the initial crack spacing would not change with changing values for  $\lambda$  and  $\alpha$ . The interplay of the geometric and physical influences might be important, however, and its investigation is one of the main aspects in this thesis. Therefore, the theses of Ecker [4] and Leindl [7] discussed in the previous section are implemented and enhanced in the course of this thesis.

Another major issue are parametric studies concerning material parameters. Parametric studies have already been carried out in the previous work [4, 7] but the design concepts of the models developed for this thesis enable the user to perform them on a larger scale. The common problems complexity and time are attacked with a high degree of automation and performance optimization.

For this purpose the crack model has to be changed to a script based model. The commercial FE simulation package Abaqus [11] used in this thesis offers a scripting interface for Abaqus CAE, the environment for designing, submitting, monitoring and evaluating complex engineering problems. This environment is written in Python and the interface enables the user to employ the full FE simulation capabilities of Abaqus CAE combined with the scripting language Python. This way the topics automation and parameterization can be treated in completely new manners.

The new design philosophy is also used to implement new concepts for the simulation of the evolution of crack arrays during service life of casting dies. On the one hand criteria are developed to manage crack array evolution in an automated manner, and on the other hand these criteria are tested to determine initial crack spacing depending on material parameters. This way crack spacing and configurations, where individual cracks will stop, do not have to be given from experimental observation on casting ingots, but result from implemented criteria. These criteria determining the evolution of crack arrays are the same throughout the parametric studies thus providing equal conditions for all variations and guaranteeing a large extent of comparability. The initial crack spacing of a crack array is a rather complex issue and studies on this topic are meant to provide a starting point for further investigations.

The ultimate goal of this thesis is to provide mathematical relations based on data fits linking material parameters with crack propagation, which can be used for material selection and material development. These data fits are supposed to reveal and comprise the interconnections between hardness,  $\lambda$  and  $\alpha$  themselves and to relieve the user of the tedious task of having to look up data from three-dimensional diagrams.

## Chapter 2

# Fundamentals of Fracture Mechanics

This section is supposed to provide the reader with the theoretical background necessary for the discussion of the approaches applied in Chapter 3 and the reasons for their selection. Furthermore, this background is necessary to identify the margins of applicability of the individual concepts behind these approaches.

The energy approach as well as the stress intensity approach to linear elastic fracture mechanics (LEFM) are essential for the development of a criterion managing automated simulation of crack configurations in crack arrays. Fatigue and crack closure as well as crack shielding and the formation of crack arrays are central issues in this thesis. They will be discussed in the following sections.

Beside papers on specific topics the following four text books shall be referred to in this section: “Fracture Mechanics - Fundamentals and Applications” [12] by Anderson provides excellent insights in general fracture mechanics, “Deformation and Fracture Mechanics of Engineering Materials” [13] by Hertzberg complements very well with [12] and gives additional information on mechanisms occurring in materials under loading and the appearance of fracture surfaces, “Fatigue of Materials” [14] by Suresh is an excellent standard work on fatigue and “Bruchmechanik - Mit einer Einführung in die Mikromechanik” [15] by Gross and Seelig further describes fracture mechanics from a continuum mechanics’ point of view.



## 2.1 Linear Elastic Fracture Mechanics

### 2.1.1 Energy Approach

In 1920 Griffith published a quantitative connection between fracture stress and flaw size (Equation 2.1) based on an energy balance derived from the first law of thermodynamics. This approach, however, is only applicable to ideally brittle solids because it assumes that the surface energy is the only factor fracture work is expended for. Nevertheless, the mathematical concepts are fundamental and provide a basis for all future work [12].

$$U - U_0 = -\frac{\pi\sigma^2 a^2 t}{E} + 4at\gamma_s \quad (2.1)$$

Griffith's derivation of Equation 2.1 is based on an elastic infinite plate under stress.  $U$  and  $U_0$  refer to the potential energy with and without a crack, respectively. The term  $-\frac{\pi\sigma^2 a^2 t}{E}$  describes the decrease in elastic strain energy of a cracked plate with increasing crack length  $a$ .  $E$  refers to the Young's modulus and  $\sigma$  to the stress. He stated that if a crack is introduced into the plate, elastic strain energy would be released accompanied by an increase in surface energy  $\gamma$ . Griffith balanced these two factors and concluded that an existing crack would propagate if the additional surface energy required was supplied by the system. The surface energy arises from the fact that neighboring atoms on a surface cannot reach the equilibrium configuration of the atoms within the solid. Thus, energy must be provided to create a surface. The term  $4at\gamma_s$  refers to this energy and is simply the product of the total crack surface area and the specific surface energy  $\gamma_s$ <sup>1</sup>. The condition of equilibrium can be obtained by minimizing Equation 2.1 with respect to the crack length [13]. The rewritten and common forms of the equilibrium condition for plane stress and plane strain conditions, respectively, are given by:

$$\text{plane stress: } \sigma = \sqrt{\frac{2E\gamma_s}{\pi a}} \quad \text{plane strain: } \sigma = \sqrt{\frac{2E\gamma_s}{\pi a(1-\nu^2)}} \quad (2.2)$$

$\nu$  refers to Poisson's ratio.

In order to utilize this criterion it is necessary to know what happens if a system satisfies the equilibrium condition. It is clear that a crack would grow if  $\sigma$  exceeds the value given

---

<sup>1</sup>Griffith based his derivation on a elliptical through thickness crack. The area of the flank is therefore  $2 * (2at)$

by the equations above because more energy is released as is needed for the formation of the surface. Now the question is whether the crack would continue growing or whether it would stop. The surface energy increases linearly with crack length, so this question can be answered by regarding the dependence of the release of elastic strain energy on the crack length. From Equation 2.1 it is evident that the decrease is not linear but quadratic. This means that the crack propagates right through the plate once the equilibrium condition is satisfied. This is also apparent from the second derivative of 2.1. Equation 2.3 has a negative sign and hence  $\frac{\partial U}{\partial a}$  will decrease with a [13]. The method Griffith used for analyzing the equilibrium in an infinite plate can be applied to real components and loads.

$$\frac{\partial^2 U}{\partial a^2} = -\frac{2\pi\sigma^2 t}{E} \quad (2.3)$$

As mentioned above, the Griffith relation is only applicable to ideal brittle material since it does not take into account the energy required for plastic flow at the crack tip. Furthermore Griffith assumed a very sharp crack. Because the crack tip radius must also be considered the Griffith relation is only a necessary condition for failure [12, 13].

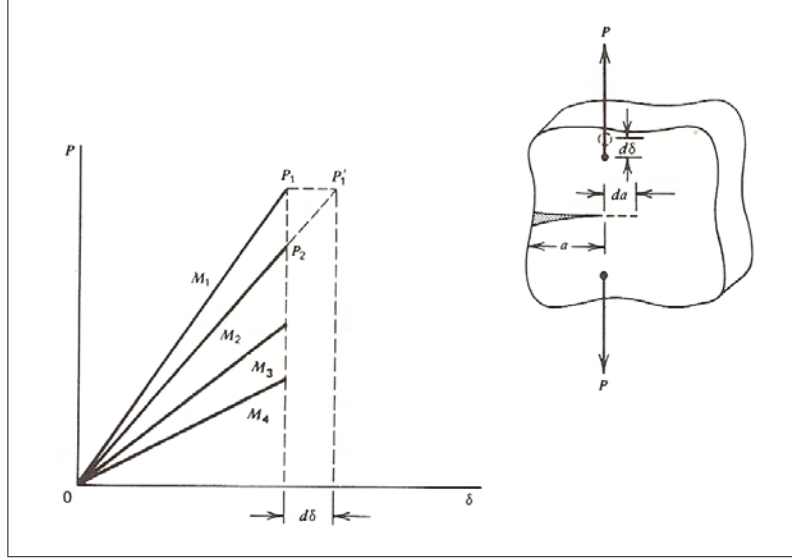
Fracture mechanics started to evolve from a scientific peculiarity to a major engineering discipline in World War II and the following decade. This change in attitude towards fracture mechanics became necessary because a significant number of vessels of the US Liberty ship program became inoperable or even broke due to cracks propagating in the newly designed hulls of the ships [12].

After the war the members of a research group at the Naval Research Laboratory in Washington D.C. became the pioneers of fracture mechanics of that time. This group was led by G. R. Irwin whose first major contribution was an extension of the Griffith approach to metals which he achieved by adding the energy dissipated by plastic flow to the energy balance, as already discussed above. This breakthrough, however, Irwin had to share with Orowan who independently came up with a similar modification.

In 1948 Irwin published an alternative concept that is easier to use in engineering [16]. The concept is called energy release rate concept and it is based on the Griffith theory. In contrast to Griffith, however, Irwin used the energy source term  $\frac{\partial U}{\partial a}$  designated by  $G$ , instead of energy sink terms, like  $\gamma_s$ . Eventually, Irwin formulated Equation 2.4 which is regarded as one of the most essential relations in the literature of fracture mechanics [12, 13].

$$\sigma = \sqrt{\frac{EG}{\pi a}} \quad (2.4)$$

In case of an elastically loaded plate (Figure 2.1) containing a crack of length  $a$ , the



**Figure 2.1:** Load-displacement curves for increasing crack lengths and strain and stress control, respectively [13].

stored strain energy can be calculated according to Equation 2.5 [12, 13].  $V$  is the stored strain energy,  $P$  the applied load and  $M$  refers to the body stiffness  $\frac{P}{\delta}$ , where  $\delta$  is the load displacement.  $M$  is a function of the crack length as depicted in Figure 2.1. In case of a fixed grip condition (displacement control), also  $P$  will vary with the crack length.

$$V = \frac{1}{2} \frac{P^2}{M(a)} \quad (2.5)$$

For a crack to propagate, energy from work done by external forces and the release of strain energy is necessary to make up for the creation of additional surface area. Consequently, the energy release rate can be formulated as Equation 2.6. After some mathematical considerations Equation 2.7 can be found. If  $G$  reaches a critical value  $G_c$ , fracture occurs.

$$G = \frac{\partial U}{\partial a} = P \frac{d\delta}{da} - \frac{dV}{da} \quad (2.6)$$

$$G_c = \frac{P_{max}^2}{2} \frac{\partial(\frac{1}{M})}{\partial a} \quad (2.7)$$

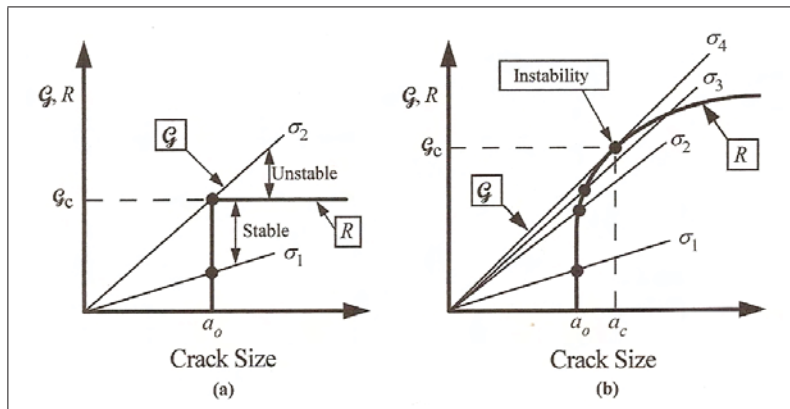
It should be considered that Equation 2.7 is valid for load control and displacement control

but that  $G$  has opposite signs (Equation 2.8). This is a consequence of the increase in strain energy due to the external force in load control [13, 17].

$$\left(\frac{\partial U}{\partial a}\right)_\delta = -\left(\frac{\partial U}{\partial a}\right)_P \quad (2.8)$$

Figure 2.2 depicts  $G$  curves and  $R$  curves for different crack lengths, where  $R$  refers to the crack resistance.  $R$  is equal to the critical energy release rate  $G_c$ , but only in case of a flat  $R$  curve, like 2.2 (a), an unambiguous value of  $G_c$  can be defined. Only in this case the crack resistance should be designated as  $G_c$ , in order to avoid confusion [17].

In case (a) the crack is stable at any stress below  $\sigma_2$  and does not grow. The relation between  $\sigma$ ,  $G$  and  $a$  is given by Equation 2.4. At  $\sigma_2$   $G_c$  is reached and the crack starts to propagate. Since, in contrast to  $R$ ,  $G$  rises continually the crack does not stop any more and thus crack growth is unstable. In case (b) the crack grows at  $\sigma_2$  by a specific increment but stops because  $R$  increases with crack length. Instability is not reached until  $\sigma_4$ , where the  $G$  curve becomes a tangent to the  $R$  curve. This point is connected to a critical crack length  $a_c$  [12]. The shape of the  $G$  curves depends on the sample geometry and the load condition. Provided that the crack length is very small compared to the sample size, the  $G$  curve will be linear. If  $a$  comes close to the magnitude of the sample size the  $G$  curve will show a positive curvature in case of load control and a negative curvature for displacement control [17]. The shape of the  $R$  curve depends on material behavior, stress state and geometry. In the present work a criterion to determine initial crack spacings as



**Figure 2.2:** Schematic driving force vs.  $R$  curve diagrams (a) flat  $R$  curve (b) rising  $R$  curve [12].

well as a criterion controlling the evolution of crack arrays were sought. Attempting to find energy based criteria seemed to be promising but was given up since in the fatigue community, energy has not yet established as quantity to describe crack propagation. To

find proper relations would have been too much work and therefore criteria based on the stress intensity approach were used. Nevertheless, the strain energy is still an appropriate means to discuss fatigue crack propagation, as done in Chapter 3.

## 2.1.2 Stress Intensity Approach

Irwin's energy release rate concept added up perfectly well with the findings of Westergaard, who showed that stresses and displacements near the crack tip can be described by a single parameter related to the energy release rate. This characterizing parameter became known as stress intensity factor  $K$ . Its application made it possible to describe crack growth with a scalar quantity [12, 14].

In a polar coordinate system with the origin at the crack tip the stress field in any linear elastic cracked body is given by Equation 2.9.  $\sigma_{ij}$  refers to the stress tensor,  $r$  and  $\theta$  originate from the polar coordinate system and  $f_{ij}$  is a dimensional function. For the high order terms  $A_m$  is the amplitude and  $g_{ij}$  is a dimensional function of the  $m^{th}$  term. Since the leading term approaches infinity as  $r$  approaches 0, whereas the high order terms remain finite or approach 0, the stress near the crack tip varies with  $1/\sqrt{r}$  independently from the high order terms.

$$\sigma_{ij} = \frac{K}{\sqrt{2\pi r}} f_{ij}(\theta) + \sum_{m=0}^{\infty} A_m r^{\frac{m}{2}} g_{ij}^m(\theta) \quad (2.9)$$

The singular stresses in the crack plane for Mode I loading are given by Equation 2.10. The validity of the singular field is restricted to a specific range of  $r$ . The upper boundary is found where the singularity dominated zone ends. At large distances from the crack tip the stress should approach  $\sigma_{\infty}$ , however, the singular term  $\sigma$  decreases to zero. Thus, the singularity dominated zone extends to the point where the contributions of the high order terms become significant. It is clear that there must be a lower limit for  $r$ , since close at the crack tip stresses given by Equation 2.10 are larger than any material could endure. Besides, plastic flow is present as well as a small process zone where blunting and micromechanical phenomena like the growth and coalescence of voids occur [15].

$$\sigma_{ij} = \frac{K_I}{\sqrt{2\pi r}} \quad (2.10)$$

Within the singularity dominated zone stress, strain and displacement can be calculated as functions of  $r$  and  $\theta$ . Thus, this so called single parameter approach is worth an entry in the list of the most important concepts of fracture mechanics. Nevertheless, this

concept is not useful, unless  $K$  can be calculated from the remote load and crack length. Fortunately, this relation can be found in the literature for different sample geometries and all of them can be related to a through thickness crack in a solid. This is possible by making use of a correction factor. The general expression relating  $K$ ,  $\sigma$  and  $a$  is given by Equation 2.11. Similar to the energy release rate a critical value for  $K$  can be found which is called fracture toughness  $K_c$ .

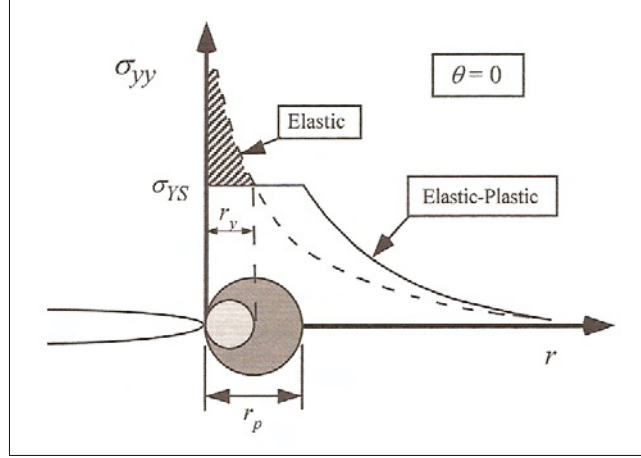
$$K = Y\sigma\sqrt{\pi a} \quad (2.11)$$

Equation 2.11 carries significant meaning for the design of components.  $K_c$  is a matter of material selection,  $\sigma$  is due to the design and loading and  $a$  could refer to a maximum allowable flaw size or a minimum detectable flaw size. If two parameters are given, the third one can be calculated. On the condition that  $K_c$  and  $\sigma$  are given, the maximum allowable flaw size can be determined. In case a minimum detectable flaw size is given by the technology used for inspection and with  $K_c$  given, it would be easy to find the critical stress  $\sigma_c$  [13]. In casting dies no external loading is present since stress fields result from temperature gradients and residual stresses. This is the reason why some aspects of LEFM need to be treated in different ways as will be discussed in Chapter 3.

Equation 2.11 has a further meaning, i.e. it is a link between the energy and stress based approaches. Both  $G$  and  $K$  can be expressed in terms of the remote load  $\sigma$ , leading to Equation 2.12 in case of a through thickness crack under plane strain.

$$K = \sqrt{\frac{EG}{(1 - \nu^2)}} \quad (2.12)$$

As mentioned above, LEFM predicts infinite stresses at the crack tip, which contradicts the observation of a finite crack tip radius and the occurrence of plasticity in metals and crazing in polymers [12]. Since LEFM becomes increasingly inaccurate the larger the inelastic zone grows, simple corrections were made available. Figure 2.3 schematically shows the Irwin approach. At the position where the stress distribution exceeds  $\sigma_y$  it will be truncated. Because equilibrium must be sustained, the stresses must be redistributed, as depicted in Figure 2.3. A first order estimate of the plastic zone size without consideration of stress redistribution was found according to Equation 2.13. A second order estimate including the redistribution gives a plastic zone size  $r_p$  exactly twice as large as predicted by Equation 2.13. These expressions were derived for plane stress conditions, so far. In case of plane strain  $r_p$  is only a third of this size due to the triaxial stress state [12, 13].



**Figure 2.3:** Correction of the plastic zone size. The presence of plasticity makes the crack behave as if it were longer. Thus, the first order estimate  $r_y$  on the plastic zone size has to be added to the actual crack length [12].

$$r_y = \frac{1}{2\pi} \frac{K_I^2}{\sigma_{ys}} \quad (2.13)$$

The plastic zone makes the material behave as if the crack were longer than its actual length. It is possible to account for this behavior by defining an effective crack length that is the sum of the actual crack length and the first order estimate  $r_y$ .  $K_{eff}$  will therefore be always larger than  $K_{applied}$ , but this difference will be very small under low stress conditions.

$$a_{eff} = a + r_y \quad (2.14)$$

Equation 2.11 can be utilized to determine  $K_{eff}$  (Equation 2.15). Since  $Y$  is dependent on crack length, an iterative solution is necessary. From a first estimate of  $K$ , without a zone correction,  $a_{eff}$  can be determined and hence  $K_{eff}$ . This procedure has to be repeated until a reasonable convergence is reached [13].

$$K_{eff} \approx Y\left(\frac{a + r_y}{W}\right) \sigma \sqrt{a + r_y} \quad (2.15)$$

Two alternative concepts of fracture analysis were found so far, namely the “Energy Criterion” and the “Stress Intensity Approach”. However, both concepts are only applicable to materials that obey Hooke’s law, apart from a limited correction for the K-approach. If

nonlinear material behavior is not limited to a small region at the crack tip, well confined in the singularity dominated zone, other concepts need to be used instead [12]. Thus, in many cases  $K$  does not describe the stress and deformation field around the crack tip and it does not have a further meaning except proportionality to the remote loading via Equation 2.11 [18].

## 2.2 Elastic Plastic Fracture Mechanics

Researchers and Engineers started to make use of the new concepts. A series of fatal accidents with Comet jet aircrafts in the 1950s were due to fatigue cracks reaching a critical size [12]. These incidents were analyzed by means of fracture mechanics and on the basis of the obtained knowledge design rules were developed still used in modern aircraft technology. The main concepts to handle fatigue are based on LEFM will be discussed in Section 2.3. Another example of early application of fracture mechanics is the prediction of the bursting behavior of large steam turbine rotors at General Electric. After LEFM had widely been accepted, a new period began in the 1960s, when focus was turned toward crack tip plasticity. Wells tried to apply LEFM to low and medium strength steels but had to abandon his efforts because these materials were too ductile. Nevertheless, he found a new parameter used extensively today. He observed that the crack faces move apart with plastic deformation, which led to the introduction of the crack tip opening displacement (CTOD) giving rise to to the development of elastic plastic fracture mechanics (EPFM).

### 2.2.1 CTOD

The CTOD results from plastic deformation at the crack tip and corresponds to the part of the Burgers vector normal to the crack flank, accumulated for all dislocations emitted from the crack tip. It can be calculated according to Equation 2.16 [18]

$$CTOD = 2\sin\alpha \int_0^{\omega} B(r)dr \quad (2.16)$$

The origin of the factor 2 lies in the symmetry of the displacement and  $\sin\alpha$  corresponds to the opening part of the Burgers vector. Based on this relation the CTOD can be linked to other parameters. The derivations and their results, however, depend on the model

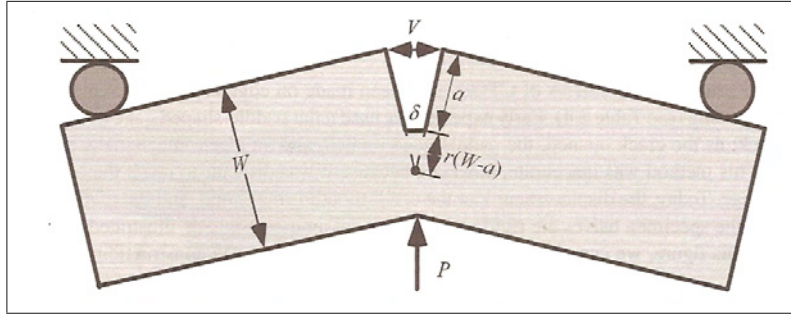


chosen and the assumptions made. In [18] a useful relation was found.

$$CTOD = 0.26 \frac{K^2}{\sigma_y E} \quad (2.17)$$

However, there are several alternative definitions of CTOD, most commonly with focus on the applicability to experimental setups. According to the most common definitions the CTOD is the displacement at the original crack tip or the displacement at the intersection of a 90° vertex with the crack flanks [12]. For laboratory measurements the most common specimens in use are edge-cracked specimens in three point bending as shown in Figure 2.4. It is assumed that the specimen halves are rigid and rotate about a hinge point. Thus the displacement  $V$  at the crack mouth can be measured and used to calculate the CTOD (Equation 2.18). The rotational factor  $r$  is a dimensionless constant between 0 and 1 and defines the relative position of the apparent hinge point [12].

$$CTOD = \frac{r(W-a)V}{r(W-a) + a} \quad (2.18)$$



**Figure 2.4:** Hinge model for CTOD estimation from three-point bend specimens [12].

### 2.2.2 J-Integral

Another important parameter used in EPFM was found by Rice. The material responses of nonlinear elastic and elastic-plastic materials differ completely during unloading but their loading behavior is identical. Provided that the stresses increase monotonically the mechanical response of both materials is the same. Rice used this fact and approximated plastic deformation as nonlinear elastic and generalized the energy release rate to nonlinear materials. He proved that the nonlinear energy release rate can be expressed in terms

of a line integral encompassing the crack tip. This integral is given by 2.19 where  $w$  is the strain energy density,  $T_i$  the components of the traction vector,  $u_i$  the displacement vector components and  $ds$  the length increment along the contour  $\Gamma$ <sup>2</sup>. The traction vector defines the stresses acting at the boundaries of a free body diagram of the material within the contour. It was found that the so-called J-Integral can be regarded as a nonlinear stress intensity parameter and as an energy release rate<sup>3</sup>. Fracture occurs when J reaches a critical value  $J_c$  [19, 12].

$$J = \int_{\Gamma} (w dy - T_i \frac{\partial u}{\partial x} ds) \quad (2.19)$$

$$w = \int_0^{\epsilon_{ij}} \sigma_{ij} d\epsilon_{ij} \quad (2.20)$$

$$T_i = \sigma_{ij} n_j \quad (2.21)$$

It was shown by Hutchinson, Rice and Rosengren that J can be used to describe stresses and strains at the crack tip.  $I_n$  is an integration constant depending on the strain-hardening exponent  $n$ .  $\tilde{\sigma}_{ij}(n, \theta)$  and  $\tilde{\epsilon}_{ij}(n, \theta)$  are dimensionless functions of  $n$  and  $\theta$  [12].

$$\sigma_{ij} = \sigma_0 \left( \frac{EJ}{\alpha \sigma_0^2 I_n r} \right)^{\frac{1}{n+1}} \tilde{\sigma}_{ij}(n, \theta) \quad (2.22)$$

$$\epsilon_{ij} = \frac{\alpha \sigma_0}{E} \left( \frac{EJ}{\alpha \sigma_0^2 I_n r} \right)^{\frac{1}{n+1}} \tilde{\epsilon}_{ij}(n, \theta) \quad (2.23)$$

Both equations are called the HRR singularity (named after Hutchinson, Rice and Rosengren), whose amplitude is determined by J in a similar way as the elastic singularity is by K. In case of small scale yielding two singularity dominated zones exist. The one in the plastic zone is described by J and the stress varies with  $r^{\frac{-1}{(n+1)}}$ . The one in the elastic regime is characterized by K and stresses vary with  $\frac{1}{\sqrt{r}}$ . The singularity dominated regime

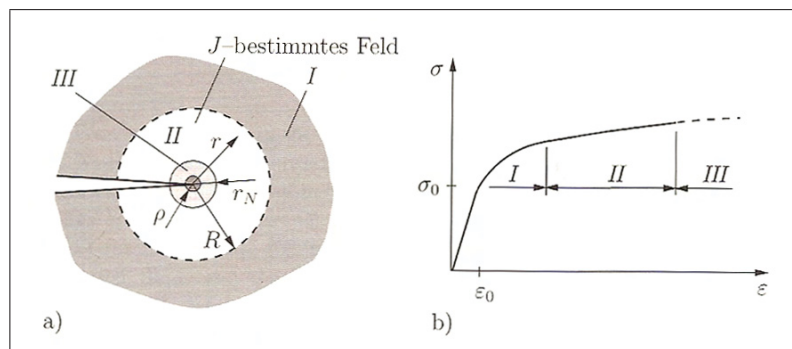
---

<sup>2</sup> $w$  and  $T_i$  are given by Equations 2.20 and 2.21 where Einstein's summation convention is assumed.

<sup>3</sup>For linear or nonlinear elastic materials J equals G.

in the plastic zone persists long after the linear elastic singularity zone has lost its validity due to crack-tip plasticity. The HRR singularity, however, neglects blunting at the crack tip.  $J$  is still a valid fracture criterion as long as there is a region around the crack tip that follows Equations 2.22 and 2.23.

Figure 2.5 depicts the  $J$  dominated zone (II) embedded in zone I where HRR is not valid any more [15]<sup>4</sup>. Zone II encloses zone III where the rules of deformation theory the  $J$  integral is based on do not apply. Zone III contains the process zone whose size is indicated by  $\rho$ . Figure 2.5 (b) shows schematically the corresponding areas in a stress strain curve. Figure 2.6 shows how the HRR field and LEFM describe the stress distribution at the crack tip.



**Figure 2.5:** (a) Area of  $J$  dominated zone and (b) corresponding sections of  $\sigma$ - $\epsilon$  diagram [15].

Later on it was shown that CTOD and  $J$ -Integral can be related to each other and that both of them are equally valid to describe crack growth unless excessive plasticity or significant crack growth occurs.

In technical applications  $J$  is usually determined by numerical methods like finite element methods. This way,  $J$  can be determined from a path independent integral, provided that the contour chosen runs through areas showing elastic responses or plastic responses according to deformation theory (no local unloading).  $J_c$  on the other hand is determined by standardized experiments [12, 13, 15].

In the 1970s Rice's findings were applied by the nuclear power industry in an effort to make use of all available means to promote the safety of nuclear power plants. Because nuclear pressure vessel steels are very tough they could not be characterized by LEFM. For this reason the  $J$ -Integral was chosen as parameter [12].

<sup>4</sup>In case of small scale yielding a  $K$  dominated zone would enclose the  $J$  dominated zone.

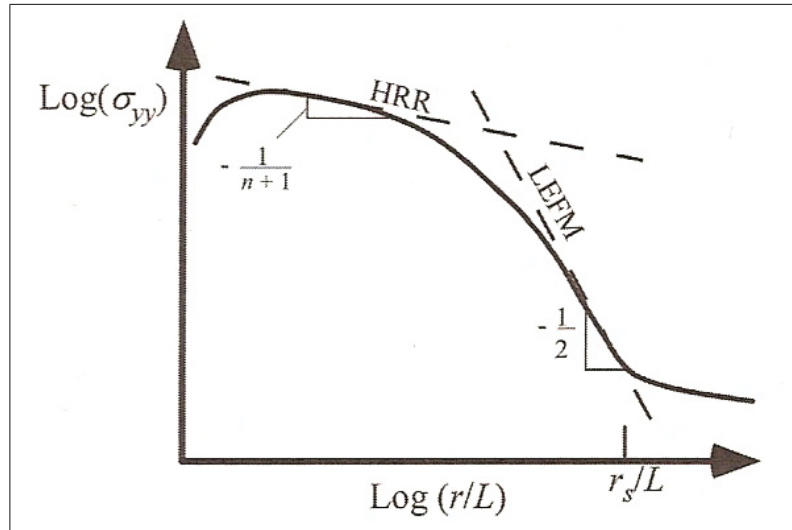
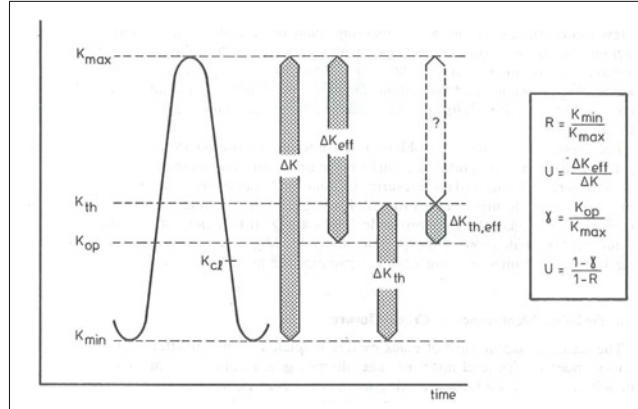


Figure 2.6: HRR field and K dominated zone in small scale yielding [12].

## 2.3 Fatigue Basics

In cases where components are subjected to loads that are not monotonic, i.e. load profiles including unloading, other concepts apply, which are summarized as fatigue concepts. Wöhler was one of the first engineers conducting systematic investigations on fatigue failure (between 1852 and 1869) and a founder of the total life approach [14]. In the 1960s Paris, Gomez and Anderson suggested the stress tolerant approach based on the observation that the increment of crack propagation per load cycle can be related to a stress intensity factor range  $\Delta K$ . Figure 2.7 shows a typical profile for a cyclic stress test and parameters characterizing this profile in terms of  $K$ . This characterization can also be done stress based, where  $\sigma_m$  is called mean stress,  $\sigma_a$  is the stress amplitude and  $\Delta\sigma$  the stress range. A very important parameter is the stress ratio, which is defined as  $R = \frac{\sigma_{min}}{\sigma_{max}}$ . It should be noted that in general this stress ratio is not equivalent to  $\frac{K_{min}}{K_{max}}$ .  $K_{min}$  and  $K_{max}$  need to be calculated by Equation 2.11 which is valid for specific cases only. In this work the stress based definition will be used.

One important issue in the course of discussing the principles of fatigue is similitude a notion providing the theoretical basis for a number of concepts in fracture mechanics. It says that stresses and deformations around the crack tip can be characterized by a single parameter, like the stress intensity factor. In case of monotonic loading this hypothesis is well established and it is true that specimens of different sizes and different crack geometries, exposed to the same loading history, will exhibit the same stress and strain distributions at the crack tip. Under specific circumstances this concept can be applied



**Figure 2.7:** Overview of parameters used to characterize cyclic loading in terms of stress intensity [20].

for fatigue and thus fatigue crack propagation can be related to  $\Delta K$  and  $K_{max}$  [21]. The Damage-Tolerant Approach which is commonly used to handle fatigue is based on the concept on similitude and thus it is important to keep in mind that its validity may be limited. There are two additional concepts, the stress and strain life approach, which are combined and referred to as the total life approach. They are important for specific applications and will be discussed briefly. Some additional views on fatigue are discussed in Section 2.6.

### 2.3.1 Damage-Tolerant Approach

In this approach fatigue life is defined as the number of cycles needed for a preexisting crack to reach a critical size [14]. This concept is counted among the most successful in fracture mechanics and based on the concepts discussed in Chapters 2.1 and 2.2.

It relates the stress intensity factor range<sup>5</sup>  $\Delta K$  ( $K_{max} - K_{min}$ ) and/or  $K_{max}$  to the crack propagation rate  $\frac{da}{dN}$ , where for constant R ratios  $K_{max}$  and  $\Delta K$  are related to each other according to Equation 2.24 [22].

$$K_{max} = \frac{\Delta K}{1 - R} \quad (2.24)$$

Subsequently, the well known Paris equation was derived. C and m are experimentally obtained scaling factors which depend on microstructure, environment, temperature and

<sup>5</sup>For tensile fatigue  $\Delta K$  refers to Mode I loading but equal concepts exist for Mode II and III.

stress ratio [14].

$$\frac{da}{dN} = C(\Delta K)^m \quad (2.25)$$

Stable fatigue growth occurs at  $K_{max}$  significantly lower than  $K_{Ic}$ , and the onset of crack growth in ductile metals usually begins at a fraction of this value [14]. In case environmental conditions are kept constant, crack growth is still influenced by  $\Delta K$  and  $R$ , which lead to the idea to characterize crack growth by these two parameters. In Equation 2.24  $R$  can also be expressed in terms of  $K_{max}$ . In the present case crack growth is characterized by  $K_{max}$ , which is computed by FE software, and the stress ratio.  $K_{max}$  is transformed to  $\Delta K_{eff}$  by means of  $R$  and an empirical relation discussed in Section 3.3.2. The parameters  $C$  and  $m$  in Equation 2.25 are determined by the function  $f$ ,  $g$  or  $h$ .

$$\frac{da}{dN} = f(\Delta K, R) = g(\Delta K, K_{max}) = h(K_{max}, R) \quad (2.26)$$

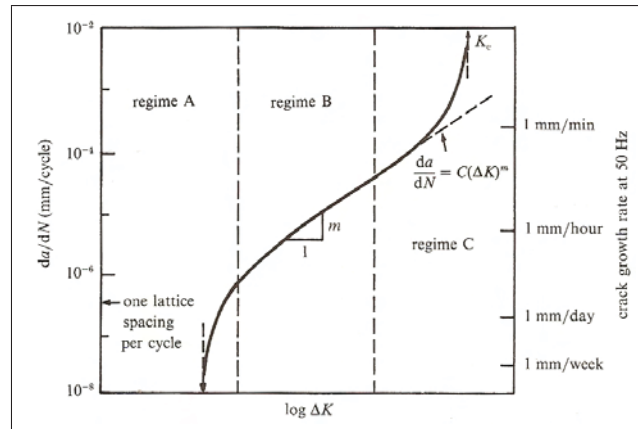
According to standard texts like [14], Paris equation can be utilized to estimate fatigue life or the number of cycles to reach a specific crack length quite easily. By considering 2.27, Paris relation can be transformed to Equation 2.28. After integration and reordering, the desired relation for fatigue life is obtained (Equation 2.29). It can be applied in different manners, typically for the calculation of load cycles until a critical crack size is reached. In this case  $a_f$  refers to the critical crack length at which catastrophic failure occurs or would occur based on estimates (i.e. from Equation 2.27) or on analyses of failed components. In cases where it is important to know how many cycles it takes for a crack to propagate by a specific increment, the corresponding crack extension can be used instead of  $a_f$ . Values for  $a_0$  are usually chosen to match the detectable flaw size during inspection or flaw sizes observed in components. The relation is valid, unless  $m = 2$ . If  $m = 2$  a less simplified form of Equation 2.29 must be used. In cases where  $\Delta K$  cannot be calculated simply from the remote load, a different procedure has to be chosen. This is the case for the simulation of crack array evolution in aluminum pressure die casting molds. The appropriate method will be discussed in Section 3.3.3.

$$\Delta K = Y \Delta \sigma \sqrt{\pi a} \quad (2.27)$$

$$CY^m (\Delta \sigma)^m \pi^{\frac{m}{2}} \int_0^{N_f} dN = \int_{a_0}^{a_f} \frac{da}{a^{\frac{m}{2}}} \quad (2.28)$$

$$N_f = \frac{1}{CY^2(\Delta\sigma)^2\pi} \ln \frac{a_f}{a_0} \quad (2.29)$$

It is generally agreed that the relation between crack propagation and  $\Delta K$  is characterized by three different regimes (A, B, C or I, II, III), as depicted in Figure 2.8. Regime III is associated with rapid crack growth, tearing or static fracture modes. Regime II, is the linear region where Paris equation is assumed to describe crack growth. In regime I crack growth is very slow and its lower border is set by  $\Delta K_{th}$ , a threshold value below which cracks are thought not to grow [22].  $\Delta K_{th}$  is also called fatigue threshold or fatigue propagation threshold and should not be confused with the fatigue limit used in the total life approach discussed in Section 2.3.2 [23]. Both thresholds are dependent on loading cycle parameters and environment and tend to decrease with rising load ratios.

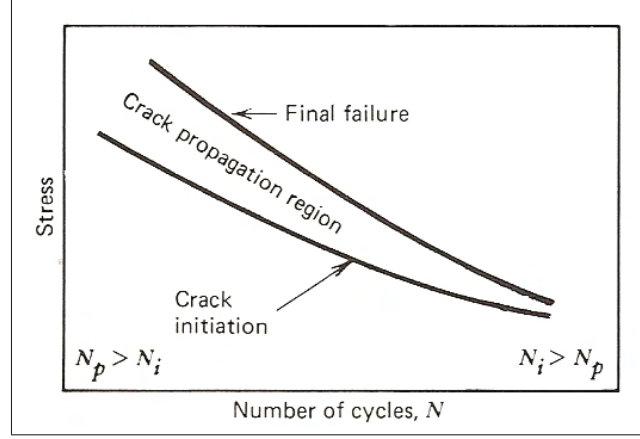


**Figure 2.8:** Regimes of stable fatigue crack propagation. Regime A: Near threshold fatigue, Regime B: Crack growth according to Paris equation, Regime C: Rapid crack growth. [14].

### 2.3.2 Total Life Approaches

In the stress and strain life approach fatigue life is defined as the total number of cycles to initiate fatigue cracks and to propagate the dominant fatigue crack to final failure. This definition is significantly different to the one in Section 2.3.1. The approach was introduced by Wöhler and led to the concept of a fatigue limit which defines a stress amplitude below which a nominally defect free material is expected to have an infinite fatigue life. Wöhler diagrams, like the one schematically shown in Figure 2.9 are used to display the fatigue life in dependence of stress amplitude. At larger stress levels the

crack propagation has a greater importance for service life compared to lower levels where crack initiation is the predominant factor. In the lower portion of the Wöhler curves a



**Figure 2.9:** Schematic of stress based Wöhler curve. The larger the stress level gets the more important becomes the crack propagation with regard to service life. At lower stress levels the impact of crack initiation is predominant [13].

linear relationship is observed. Basquin derived an expression which relates the stress amplitude, in a fully reversed ( $R=0$ ), constant amplitude fatigue test to the number of load reversals to failure  $2N_f$ .  $\sigma'_f$  is the fatigue strength coefficient and  $b$  is the fatigue strength exponent (Basquin exponent) [14]. This relation describes fatigue life in high cycle fatigue (HCF), where corresponding strain amplitudes are in the elastic regime.

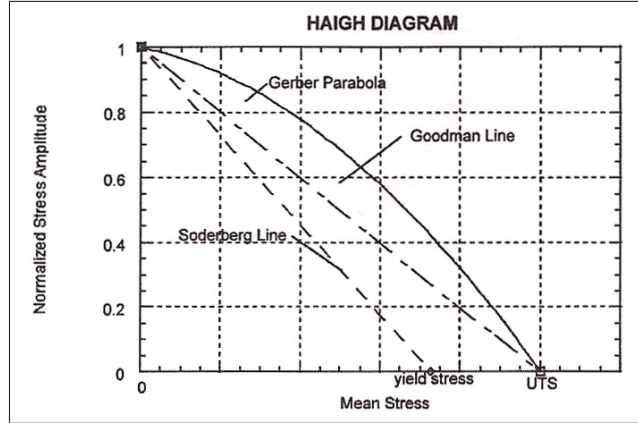
$$\sigma_a = \sigma'_f (2N_f)^b \quad (2.30)$$

Wöhler diagrams, however, are strongly dependent on the mean stress. This dependence is represented in of Haigh diagrams (Figure 2.10). In such diagrams curves are plotted in coordinates of mean stress vs. cyclic stress amplitude in order to display conditions for “infinite fatigue life” which, of course, cannot be treated as such. Figure 2.10 is also called fatigue limit diagram and the stress at the point where the mean stress is zero is defined as the fatigue limit ( $R=-1$ ) [23]. An alternative kind of the Haigh diagram which includes additional information like the stress ratio is also commonly used.

Coffin and Manson proposed an alternative characterization of fatigue life based on the plastic strain amplitude. They found a linear relationship for the plastic regime (low cycle fatigue, LCF) like Basquin did for the elastic regime [14].

$$\frac{\Delta \epsilon_p}{2} = \epsilon'_f (2N_f)^c \quad (2.31)$$





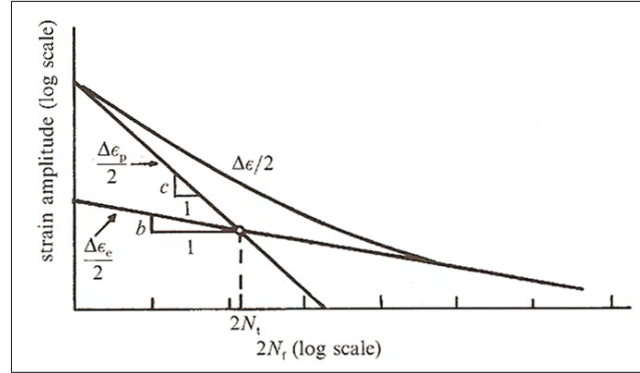
**Figure 2.10:** One possible form of a Haigh Diagram. The curve found by Soderberg gives a conservative estimate for most engineering alloys. The Goodman relation gives very good results for brittle metals and the Gerber relation is appropriate for ductile metals and tensile mean stresses [23].

$\epsilon'_f$  is the fatigue ductility coefficient and  $c$  refers to the fatigue ductility exponent. The strain amplitude in a constant strain amplitude test can be expressed as the sum of elastic and plastic strain amplitude. Thus, the findings of Basquin, Manson and Coffin complement each other and can be used to describe fatigue life in the strain life approach. Equation 2.32 is the combination of the relations found previously, where the first term describes HCF (elastic regime) and the second one LCF (plastic regime). Figure 2.11 depicts a strain based Wöhler diagram with both approximations.

$$\frac{\Delta\epsilon_p}{2} = \frac{\sigma'_f}{E}(2N_f)^b + \epsilon'_f(2N_f)^c \quad (2.32)$$

## 2.4 Near Threshold Fatigue

Near threshold fatigue takes place at and below the border between regimes I and II at propagation rates less than  $10^{-6} \frac{mm}{cycle}$  [23]. Cracks commonly grow in this range after initiation in HCF. It is important to distinguish between fatigue growth of cracks in the near threshold regime which propagate with less speed as expected from Paris equation and fatigue growth of physically small cracks which grow faster than long cracks. In regime I cracks are always small but in terms of crack propagation these cracks are usually long. More information on the difference between long and small cracks is given in Section 2.6.



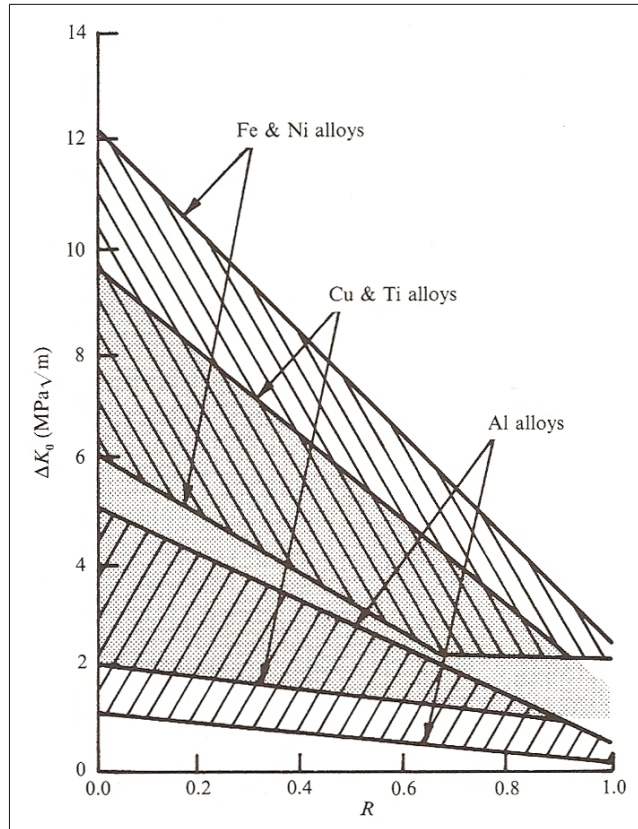
**Figure 2.11:** Strain based Wöhler curve illustrating the findings of Basquin, Manson and Coffin [14].

Fatigue thresholds and limits, however, are not absolute borders, below which  $\frac{da}{dN}$  drops to zero. If the method used for determining crack lengths in a crack propagation experiment is accurate to 0.1 mm and no crack propagation is detected after  $10^7$  cycles, the fatigue threshold is considered to have been reached [14]. It should also be considered that, as mentioned above, the fatigue threshold depends on several factors like microstructure, stress ratio, environment and crack size. Figure 2.12 shows typical ranges of  $\Delta K_{th}$  for various engineering alloys. At low R values differences in microstructure and between materials are more significant than at high R values. For Fe and Ni alloys a plateau is reached at high load ratios [14]. Due to limited plasticity in near threshold fatigue, LEFM is the appropriate tool to describe crack propagation [23]. In order to describe crack growth in this regime properly, an adapted version of Paris equation was suggested by Elber.

$$\frac{da}{dN} = C(\Delta K - \Delta K_{th})^m \quad (2.33)$$

In case of short cracks, the use of LEFM to describe crack propagation is debated [23]. It is discussed whether similitude breaks down for short cracks or not. More information on this issue can be found in Section 2.6.

Nevertheless, Equation 2.33 can be used to describe crack growth of long cracks in the near threshold regime, provided that the fatigue threshold is known.



**Figure 2.12:** Typical ranges of fatigue threshold for some engineering alloys. Fe and Ni show a plateau which corresponds to an intrinsic threshold  $\Delta K_{th}$  [14].

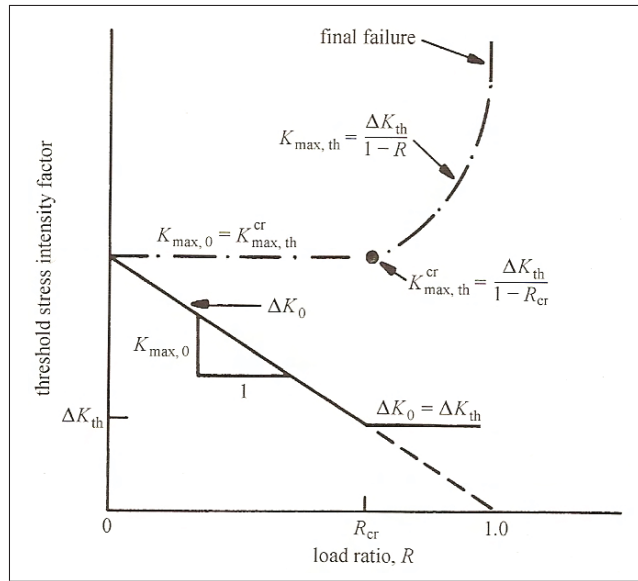
## 2.5 The Fatigue Threshold

The most common approach to consider the changes in the fatigue threshold is to implement the findings of research carried out on crack closure, which will be discussed in Section 2.7. In this approach, the whole crack propagation curve is shifted to match up with the curves of other load ratios. This way crack propagation is transformed into a function of an effective intensity factor range  $\Delta K_{eff}$ . In this consideration only one threshold exists, i.e. the intrinsic threshold  $\Delta K_{th}$ .

Fe and Ni show a plateau in Figure 2.12. For these metals the dependence of the fatigue threshold on  $R$  can be expressed in terms of the stress intensity factor range  $\Delta K_0$  and the maximum value of stress the intensity factor  $K_{max,0}$  [24]. Figure 2.13 schematically shows how to handle the presence of a fatigue threshold which is dependent on stress ratio with these two parameters.  $\Delta K_{th}$  in this context is the lowest value of  $\Delta K_0$ , which is constant above a critical value of stress ratio  $R_c$ . If the fatigue crack growth threshold is treated in terms of the stress intensity factor  $K_{max,0}$  it can be said that  $K_{max,0}$  is equal to a con-

stant  $K_{max,th}^{cr}$  below  $R_c$ . Above this stress ratio  $K_{max,0}$  increases with  $R$ . The shape of the curves in Figure 2.13 depends on the type of material, however, which is discussed in the paper of Vasudevan and Sadananda [25]. The fatigue threshold is therefore determined by two parameters, where one of them is always constant in the particular  $R$  range. This approach is popular in the community around Vasudevan who argues that there exists both an intrinsic threshold of  $\Delta K$  and  $K_{max}$  which determine the onset of crack growth. In this community crack closure is regarded as less important and is defined as a third but extrinsic parameter. Further differences between the concepts based upon one or two thresholds are discussed in detail in the paper of Krenn and Morris [26].

The present work follows the first approach based on  $\Delta K_{th}$  as intrinsic threshold and crack closure as extrinsic parameter. Crack propagation curves for various load ratios were shifted in order to obtain a curve for the effective driving force. The quantity received from the FE software, however, is  $K_{max}$ . Thus, both parameters were used but only  $\Delta K_{th}$  was regarded relevant for a criterion managing crack growth. The exact procedure for the present case is described in Section 3.3.2.



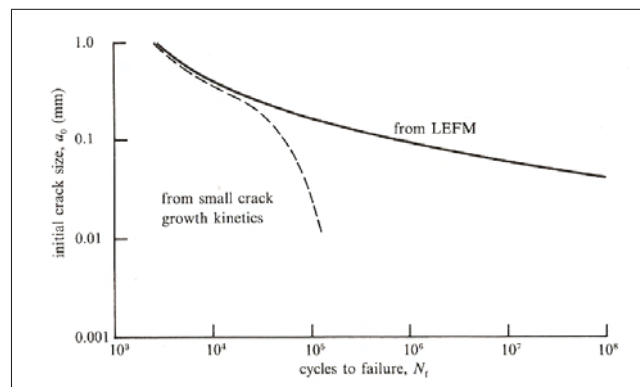
**Figure 2.13:** Schematic relation between the maximum stress intensity and the stress intensity range which illustrates the possibility to treat the onset of crack growth in both terms [14].

## 2.6 Small Fatigue Cracks

In case of long cracks under small scale yielding conditions the propagation of fatigue cracks can be described by the stress intensity factor range  $\Delta K$  and lifetime predictions can be found quite easily, at least in regime II where similitude is justified by the Paris relation. In some cases Paris equation was successfully applied to short cracks but thresholds often differ between small and long cracks as stated in [23].

However, in many investigations carried out on small crack growth, mostly with continuum approaches, it has been shown that propagation rates of small cracks can be significantly larger than for long cracks when characterized by  $\Delta K$ <sup>6</sup>. Hence, design against fatigue failure, based on standard methods, can come along with an overestimation of service life, when components containing short cracks are concerned. In this section methods will be introduced on how to identify small cracks and how to treat them with respect to fatigue life estimations.

Figure 2.14 contains estimates for life time based on LEFM and small crack growth kinetics as functions of initial flaw size, in case of a nickel-base superalloy. For initial crack sizes larger than 0.3 mm LEFM gives proper estimates whereas estimates for smaller initial sizes are nonconservative [14].



**Figure 2.14:** Number of fatigue cycles to failure based on LEFM and small crack growth kinetics. The latter one gives nonconservative results for small initial sizes [14].

First of all, however, it is essential to be able to differentiate between long and short cracks, which is not trivial, partly because of the various definitions of these terms. Understanding the differences between these definitions and their backgrounds is very important.

For the considered superalloy in the example above cracks shorter than 0.3 mm could be

<sup>6</sup>Please mind that Equation 2.33 is valid for long cracks, i. e. physically small cracks.

called small, but this is no general definition. In the literature cracks smaller than 1-3 mm are said to be small but, at least in a physical sense, small cracks are characterized by the breakdown of similitude [18]. This breakdown can be caused by various reasons and these in turn lead to various definitions of small cracks. Riemelmoser and Pippan follow in their paper the definitions of Suresh [14].

### Microstructurally Small Cracks

Cracks with dimensions in the range of microstructural characteristics, like grain size or interlamellar spacing, are subject to environments formed by elastic and plastic anisotropies and residual stresses characteristic for this length scale. The microstructural influences diminish for cracks exceeding the characteristic length scale of the microstructure by the factor 5 to 10.

### Mechanically Small Cracks

In case of long cracks the plastic zone is embedded within the K-dominated stress field and its size is given by Equation 2.34 according to LEFM.

$$r_p = \frac{1}{2\pi} \frac{K^2}{\sigma_y^2} \quad (2.34)$$

According to Riemelmoser and Pippan [18], the plastic zones of small cracks are much larger, in case of equal values for K, and they are not entirely situated within the K-dominated field. This effect can be discussed when regarding Figure 4.11 in Chapter 5. The simulation of crack growth starts at a crack length of 0.1 mm.  $\Delta K_{eff}$  increases at first and after reaching a maximum value decreases back to the initial value and below, i. e. the same value of  $\Delta K_{eff}$  occurs at two different crack lengths. At 0.1 mm, however, the crack might be mechanically small which will be discussed in Section 4.5. Cracks are called mechanically small as long as the deviations from the long crack solution are significant.

### Physically Small Cracks

Extrinsic effects that influence the closure level of cracks decide whether cracks are designated as physically small cracks. The dynamic stress intensity factor may be shielded due to crack flank contact behind the crack tip. Contact is made within an area of the size of the cyclic plastic zone, at least in case of plasticity induced crack closure under plane strain conditions (see Section 2.7.2). Crack closure cannot develop its full effect in cracks smaller than that area. Physically small cracks are thus cracks too small to develop

the full closure level. Because also for long cracks situations exist where the closure level differs from steady state values, the term extrinsically small crack is sometimes regarded as the more favorable one, since its meaning is clearer. In disturbed long cracks this effect is due to the loading history. Nevertheless, both cases have a physical background.

### Chemically Small Cracks

Whenever crack propagation is affected by chemo-mechanical processes, cracks can be called small relative to characteristic length scales involved in the process, like the hydrogen affected zone or the chemical transport zone behind the crack tip.

### Short Cracks

The difference between small and short cracks can be explained by the dimensions taken into account. Small cracks are small in three dimensions and form naturally. Short cracks are small in two dimensions but not in the third one (through thickness cracks). They are usually formed artificially like an incipient crack in a compact-tension-sample (CT-sample) [18]. In many contexts, however, this difference is not considered.

The regimes of different types of small cracks overlap often but not always. Microstructurally small cracks are typically small as well if mechanical and extrinsic aspects are taken into account. Mechanically small cracks usually are also extrinsically small cracks. Cracks in very coarse microstructures, however, may be mechanically short but not with respect to the microstructure.

Since experiments are commonly carried out on short cracks this term will be used in the following. The interactions of  $\sigma_y$ , crack length and loading condition are not trivial and therefore their impact on characteristic quantities, like the CTOD, cannot be described by simple relations. The border between the long and short crack regime, however, can be found easier. For various crack sizes stress intensity factors can be calculated, at which the long and short crack solutions differ by more than 10 %. The same way characteristic crack sizes can be found in case of given stress intensity factors. This approach leads to very straight border lines between the long and short crack regime. According to Riemelmoser and Pippan [18] these lines can, in case of uniaxial loading, be described by the corresponding Equations 2.35, 2.36 and 2.37.

$$K \leq \frac{\sigma_y}{10} \sqrt{\pi a} \quad (2.35)$$

$$\sigma_{yy}^{\infty} = \frac{\sigma_y}{10} \quad (2.36)$$

$$\frac{1}{\pi} \frac{K^2}{\sigma_y} \leq \frac{a}{100} \quad (2.37)$$

Equation 2.35 can be used to characterize a crack by the stress intensity factor applied. If  $K$  is larger than the expression on the right hand side of the equation, the crack is in the short crack regime. Equation 2.36 like Equation 2.37 is a reformulation of the first equation. According to Equation 2.36 a crack is short if the applied stresses exceed a tenth of  $\sigma_y$ . The equation thus illustrates the influence of the yield stress. The left hand side of Equation 2.37 corresponds approximately to the plastic zone size obtained from continuum mechanics. The plastic zone size can therefore be used as a criterion as well. Even though  $\Delta K$  does not have a justification as driving force in a mechanical and physical point of view beyond the borders given by the equations discussed above, it is still a widely used means to describe the propagation of short cracks. Therefore corrections are necessary that take into account the deviations of the short crack solutions from the long crack solutions .

Riemelmoser and Pippan provide an adapted formulation of Equation 2.17 that gives a good and also conservative approximation of the short crack solution. Equation 2.38 can be used instead of full elastic plastic analyses, if very accurate solutions are not necessary.

$$CTOD = 0.26 \frac{K^2}{E(\sigma_y - \sigma_{yy}^{\infty})} \quad (2.38)$$

The above-mentioned relations were derived for static loads. In case of cyclic loading, however, the relations can be used as well by applying Rice's scheme where  $K$  is replaced by  $\Delta K$ ,  $\sigma_y$  by  $2\sigma_y$  and  $\sigma_{yy}^{\infty}$  by  $\Delta\sigma_{yy}^{\infty}$  [18].

Equation 2.38 changes to Equation 2.39

$$\Delta CTOD_{corr} = 0.26 \frac{\Delta K^2}{E(2\sigma_y - \Delta\sigma_{yy}^{\infty})} \quad (2.39)$$

In case of general loading a further correction needs to be applied to this expression. Based on the derivations in [18] Equation 2.39 can be reformulated in order to be able to



treat this issue in terms of  $\Delta K$  only.

$$\Delta K_{corr} = \Delta K \sqrt{\frac{2\sigma_y}{2\sigma_y - \Delta\sigma_{yy}^\infty}} \quad (2.40)$$

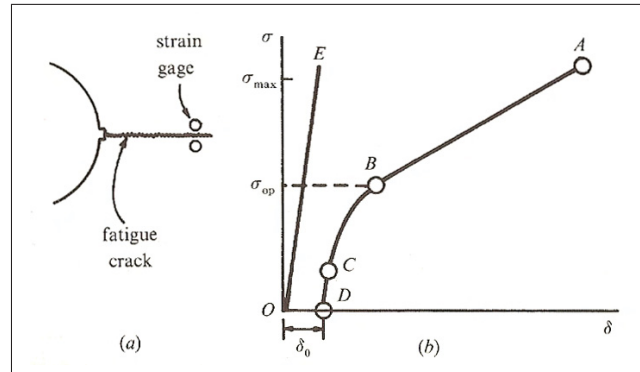
Riemelmoser and Pippan found the correction of  $K$  according to Equation 2.40 yielding adequate approximations being in good agreement with measured propagation rates and effective growth curves [18].

## 2.7 Crack Closure

### 2.7.1 Overview

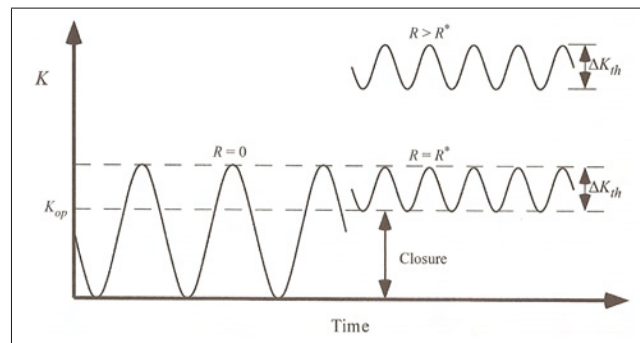
Crack closure is a crucial issue whenever Paris equation is applied for small or negative load ratios. Due to plastic deformation, roughness of the crack flanks, oxide or metal particles, or other phenomena, premature closure can occur [23]. Thus, the relative motion of the crack flanks and the range between which  $K$  is varying, seen at the crack tip, are lower than expected from the external loading. This phenomenon was first observed by Elber, who explained it with residual plastic strains in the wake of the crack. Elber observed and explained this by stress-dislocation curves [14]. Figure 2.15 shows schematically what happens when a specimen is unloaded from a stress level, where the crack is completely open. The section between points A and B is linear, which indicates that the crack flanks do not have contact. It corresponds to the stiffness of the sample with a crack of the length  $a$ . E is the stiffness of the sample without a crack. In the section between point B, where  $\sigma_{op}$  is reached and point C, contact is established. After point C the stiffness of the cracked specimen is equal to the one without crack, which means that contact is fully established and the crack is closed in that stage. This finding of a closed crack at tensile stress levels was the first evidence of the phenomenon crack closure. Elber concluded that a crack can only propagate during stages of a loading cycle where the crack flanks are separated. The procedures necessary to take crack closure into account for predictions of service life are explained in [12] and [14]. Elber's conclusions led to the definition of an effective stress range and an effective stress intensity factor range. The parameter  $\Delta K_{eff}$  has to be used in Paris equation instead of  $\Delta K$ . In the calculation of  $\Delta K_{eff}$  often  $K_{min}$  is used instead of  $K_{op}$  which is not correct, unless  $K_{min} = K_{op}$ .

$$\Delta\sigma_{eff} = \sigma_{max} - \sigma_{op}, \quad \Delta K_{eff} = K_{max} - K_{op} \quad (2.41)$$



**Figure 2.15:** Illustration of the relationship between applied stress and displacement with crack closure involved [14]. In the linear section between points A and B the crack flanks do not have contact. Between point B and point C contact is established. After point C contact is fully established.

Figure 2.16 illustrates the effect of crack closure on the threshold value of  $\Delta K$  ( $K_{op}$  is



**Figure 2.16:** Influence of  $R$  on the intrinsic threshold value  $\Delta K_{th}$ . Below  $R^*$  the apparent threshold for crack propagation is larger than the intrinsic value because of crack closure [12].

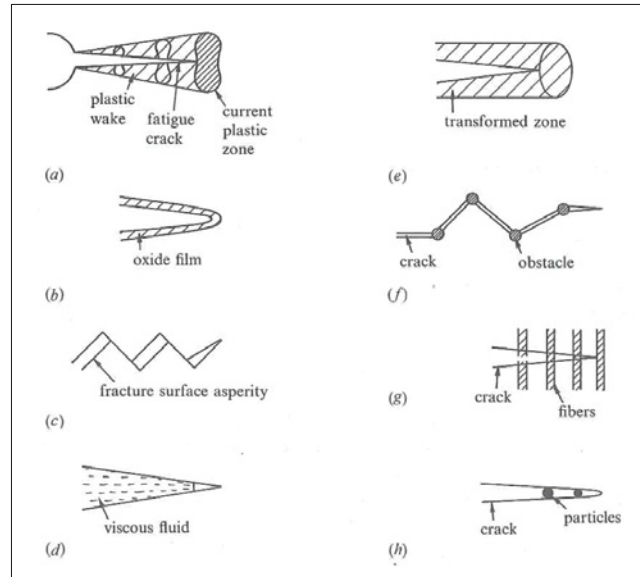
assumed to be independent of  $R$  in this example). Above the stress ratio  $R^*$  crack closure does not have an influence. The corresponding threshold value in the schematic is the intrinsic threshold  $\Delta K_{th}$ . Below  $R^*$  the intrinsic threshold is the same but from external loading an apparent threshold which is larger arises.

In the opinion of the majority of experts there exists an intrinsic threshold  $\Delta K_{th}$  which is a material property and an extrinsic threshold dependent on  $R$  due to crack closure [12]. A second, smaller group thinks that crack closure has less relevance, though. They believe that there are two intrinsic thresholds, a threshold for  $\Delta K$  and one for  $K_{max}$ .

In this thesis crack closure is regarded as significant and the applied concepts are based on the findings of the first group.

Figure 2.17 shows various mechanisms, causing crack closure and thus reduced driving

forces.

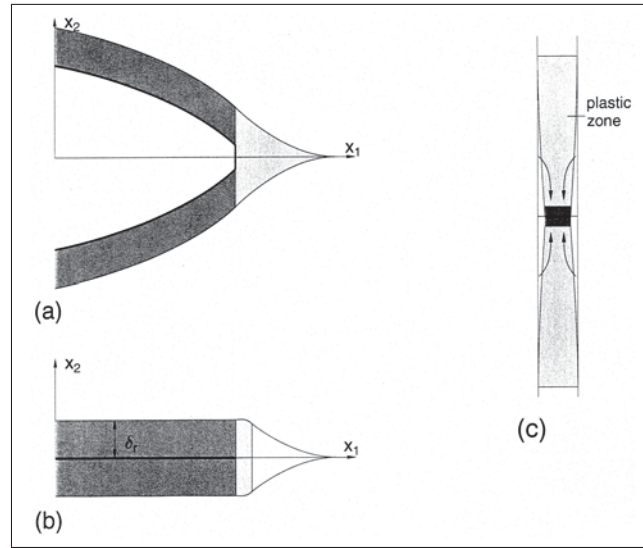


**Figure 2.17:** Different mechanisms of crack closure: (a) plasticity-induced crack closure; (b) oxide-induced crack closure; (c) roughness-induced crack closure; (d) fluid-induced crack closure; (e) transformation-induced crack closure; (f) crack deflection; (g) crack-bridging by fibers; (h) crack-bridging by particles [14].

A deeper understanding about these crack retardation mechanisms is necessary for accurate predictions of fatigue life. Particular attention should be paid to plasticity induced crack closure especially under plane strain conditions. Because plane strain prevails in the casting dies the next section will focus on this topic.

## 2.7.2 Plasticity Induced Crack Closure

According to Riemelmoser and Pippan [27] the reason for plasticity induced crack closure is extra material behind the crack tip that causes contact of the crack flanks at an earlier stage than expected. Under plane stress conditions this is a well studied phenomenon and the origin of this extra material is quite easy to explain. A plate becomes thinner in the area around the crack tip and material is transported to the flanks. This material transport takes place during each crack growth increment and the entire fatigue crack flanks are padded by the extra material. Figure 2.18 illustrates this mechanism. The mechanism of material transport under plane strain conditions is very different and at a first glance there is no possible source. It is unclear why plastic deformation at the crack tip perpendicular to crack propagation should cause material transport in propagation



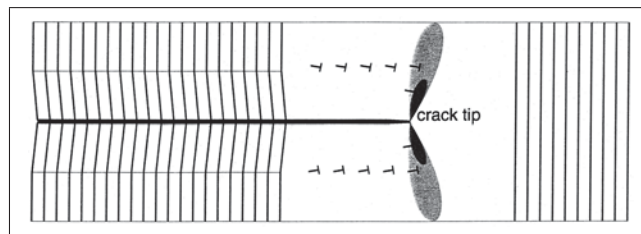
**Figure 2.18:** Schematic of plasticity induced crack closure under plane stress conditions. Extra material at the crack flanks originates from plastic out of plane flow [28].

direction. Nevertheless, there are many experiments giving evidence of plasticity induced crack closure in the interior of specimens where plane strain conditions are predominant. In [28] the mechanisms behind these observations are discussed in detail. In this paper Pippan and Riemelmoser briefly discuss the role of roughness-induced crack closure, which could explain the plastic wedge causing contact of crack flanks even under far-field tensile loads. They pointed out, however, that additional concepts are required since crack closure also arises on smooth fatigue crack surfaces and the plastic wedge appears in FE simulations which do not consider roughness.

In order to understand the mechanism discussed in [27] and [28] the effect of cyclic plastic deformation on the material in the wake of the crack needs to be understood. This can be done by having a look at the geometrically necessary dislocations in that area. Their Burgers vectors must allow both tip blunting and crack propagation. The volume in the strip behind the crack does not change, however, since plane strain conditions do not allow out of plane flow ( $\epsilon_{33}$ ) and  $\epsilon_{11}$  and thus due to volume constance  $\epsilon_{22}$  can be regarded as zero. The last statement is an assumption based on the fact that plastic deformation occurs only in a small strip. Eventually, all these arguments lead to a dislocation arrangement shown schematically in Figure 2.19. This Figure illustrates that elements near the crack flanks have been exposed to shear in the direction of crack propagation and therefore have been tilted. Similar dislocation configurations found in small-angle grain boundaries are known to rotate the crystal elastically on one side of the crystal with respect to the other. The same mechanism applies in the wake of a crack, where the areas

in the vicinity of the flanks are rotated with respect to the areas further away. Because of the rotation some material is missing on the left side, in case of the configuration shown in Figure 2.19, and an equal amount of material can be found at the crack tip. The extra material, however, travels along with the crack tip and no trace of it is left on the flanks. This is a fact important to be aware of, since it is a crucial difference to the plane stress case. In contrast to plane stress conditions not the entire fatigue crack is padded by extra material but only a small area behind the crack tip. This is the reason why this effect can hardly be observed in experimental setups.

In [27] Riemelmoser and Pippan also give an alternative explanation for the elastic material transport on the basis of continuum mechanics.

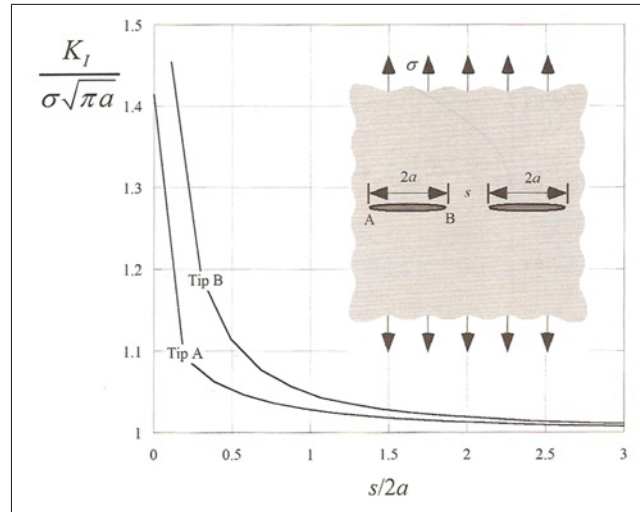


**Figure 2.19:** On the left hand side: Volume transfer to the crack tip by rotation of volume elements in the wake during crack growth under plane strain conditions; At the center: Monotonic and cyclic plastic zone; On the right hand side: Remaining ligament [28].

## 2.8 Crack Arrays

### 2.8.1 Fundamentals

At the beginning of this section the basic interaction between multiple cracks should be discussed. According to Anderson [12] there are two extremes, i. e. the amplification of stress intensity factors at the crack tips compared to the case of a single crack and conversely crack shielding leading to lower  $K_I$ . The first extreme can be observed when cracks are aligned in the same plane with propagation direction perpendicular to external forces. This configuration is shown in Figure 2.20 as well as the normalized  $K_I$  vs. normalized crack spacing. As expected, the stress intensity factors rise with decreasing crack distance. Because of the amplification of  $K$  multiple in-line cracks are a threat to components. The second case is essential in view of the crack arrays developing in casting dies. Figure 2.21 shows a configuration with parallel cracks. The stress distribution changes in such a way that the cracks shield each other and the stress intensity factors



**Figure 2.20:** Interaction of two identical coplanar through thickness cracks in an infinite plate [12].

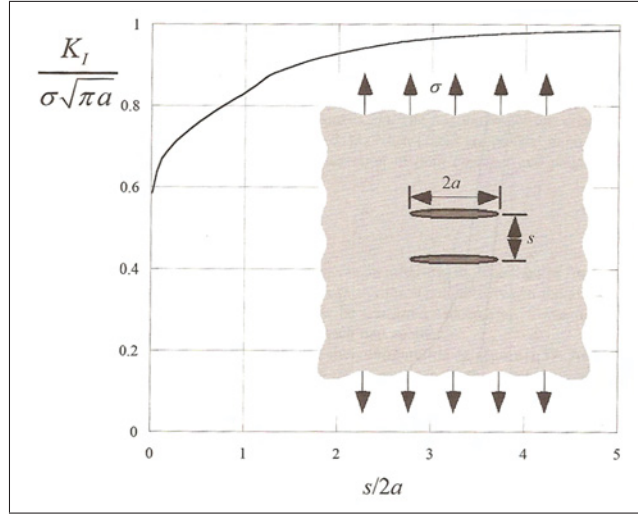
decrease with decreasing spacing. Engineers can take advantage of the shielding effect by deliberately creating stress relief by minimizing the crack spacing in the case of arrays of parallel cracks. This way more serious damage like fast propagating single cracks can be avoided.

### 2.8.2 Former Studies on Crack Arrays

The issue of interacting cracks in crack arrays was investigated in the context of different practical problems. Nemat-Nasser [29] et al. had a look at the formation of cracks in rock masses used for extracting heat. Haddar et al [30] investigated the evolution of crack networks in AISI 304L steel used for nuclear reactor components. Jenkins [31] studied crack formation in solidifying coke, which show similar patterns as in basalt columns. In their papers all of them provide theoretical approaches which can be adapted for the present study or used for the development of new concepts. The most important ones will be discussed now, beginning with the paper of Nemat-Nasser et al. which was a cornerstone for many other scientific articles.

#### Formation of crack arrays in rocks

The elastic energy stored in a material subjected to thermal contraction, like in the case of rock cooled by circulating water in geothermal plants or likewise in pressure die casting



**Figure 2.21:** Interaction of two identical parallel through thickness cracks in an infinite plate [12].

molds after spraying, is finite. This is the reason why there is always a finite number of cracks with a minimum average crack spacing at the surface. This is true for monotonic loading as well as load cycles leading to fatigue.

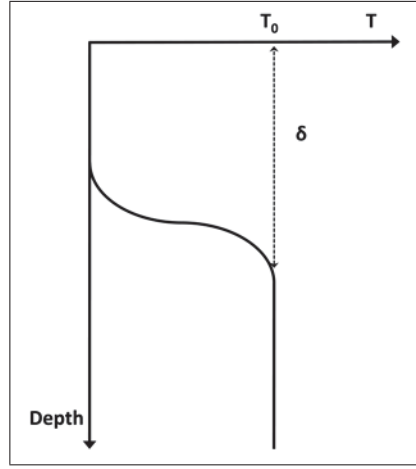
For the discussion of instabilities in the crack array a unit cell, consisting of two interacting cracks, can be regarded. In equilibrium both cracks in the unit cell are of equal length and their stress intensity factors equal  $K_c$  (2.42). The stress intensity factor of each crack depends on the lengths of both cracks and a load parameter  $\delta$ , which is also called penetration depth. It is a measure that increases with square root of time and is also a measure for the depth in which a significant temperature gradient is observed (Figure 2.22).

$$a_1 = a_2 = a \text{ and } K_1 = K_2 = K_c \quad (2.42)$$

$$K_i = K_i(a_1, a_2, \delta_0) \quad (2.43)$$

It is not sufficient to simply identify equilibrium. Moreover a criterion must be defined deciding whether this equilibrium is stable, unstable or critical. According to Nemat-Nasser [29] this issue can be treated in the following way.

Condition 2.42 is stable when the terms in 2.44 are negative. There is no possibility



**Figure 2.22:** Temperature distribution due to convection, redrawn from [29].

for either of the two cracks to grow spontaneously for any  $da_i \geq 0$ , since  $dK_i$  is always negative or equal to zero. In case of interacting cracks equilibrium becomes unstable if condition 2.45 is true. If the terms in 2.45 are equal to zero a critical state is present at which one of the cracks will stop. This is logical since, compared to the other states in which both cracks can grow or cannot grow, at this point the only way for crack growth to occur is that one of the cracks stops. In view of energy this is more obvious. At the critical state a reduction in potential energy is not possible, if both cracks advance, because elastic energy released would be entirely consumed by the newly formed crack surfaces and hence the potential energy  $U$  would stay the same. For this reason this state is critical, because for the system it does not make any difference if both cracks grow or do not grow by an increment  $\delta a$ . If only one crack advances, on the other hand, the potential energy of the system decreases because less energy is spent for the creation of new crack surfaces. The growing crack draws stored strain energy from a larger area than during the preceding increment  $\delta a$  where it had to share the energy with the second crack. Thus it will grow, at the same time decreasing  $U$ . Note that the energy released by reducing  $U$  is used to drive the crack.

Although crack propagation follows a different scheme in aluminum pressure die casting dies, since it occurs far below  $K_c$  and due to non monotonic loading, the discussion of a critical crack configuration will be picked up again in Section 3.3.2.

$$\frac{\partial K_1}{\partial a_1} = \frac{\partial K_2}{\partial a_2} < 0, \text{ stable} \quad (2.44)$$



$$\frac{\partial K_1}{\partial a_1} = \frac{\partial K_2}{\partial a_2} > 0, \text{ unstable} \quad (2.45)$$

$$\frac{\partial K_1}{\partial a_1} = \frac{\partial K_2}{\partial a_2} = 0, \text{ critical} \quad (2.46)$$

In rock masses,  $\delta$  increases with time because heat energy is extracted steadily and the temperature gradient moves deeper into the stone. Figure 2.23 schematically explains the situation. Based on the assumption that there are initially two equal cracks in the unit cell of the array, it can be said that both will grow stable until a critical point  $C_1$  is reached. One of the cracks will stop (crack one in this example) while the other one continues to propagate in a stable manner as long as  $\frac{\partial K_2}{\partial a_2}$  is negative. Between  $C_1$  and  $C_2$   $K_1$  decreases and finally reaches zero at a second critical point  $C_2$ . Since  $K$  cannot become negative, it closes and crack number two reaches a finitely longer length  $a_P$ . The same will happen to all unit cells the system consists of and other critical states will be reached, when the advancing cracks of the individual unit cells will start to interact.

As discussed in Section 2.1 there is an alternative approach to LEFM based on energy. Thus, a criterion can be developed by comparing the energy stored in the system to the energy needed for crack propagation, similar to Equation 2.1. This criterion is discussed in [29].

In the present work the symmetry of the crack arrays was used to build a FE model for the simulation of crack array evolution. This way criteria applicable to fatigue could be implemented and computation time was reduced to a minimum. Further details can be found in Chapter 3.

### Formation of Crack Arrays in Solidifying Coke Layers

Jenkins investigated crack growth in solidifying coke layers. Coke is supposed to be lumped and thus cracking is desired during industrial production. In [31] Jenkins implemented the results found by Nemat-Nasser and others in order to conduct FE simulations on crack arrays. In previous studies, however, the initial crack spacing was specified, whereas Jenkins also implemented concepts to find the optimal initial spacing of the arrays.

The approach chosen in [31] is to compute configurations of the array for a range of different values of  $a$  and  $l$  and to find those which minimize the potential energy  $U$  of the

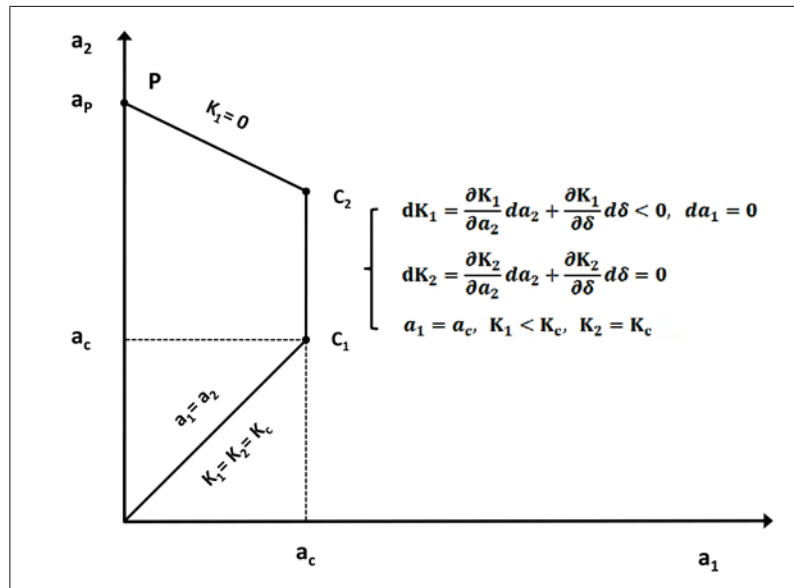


Figure 2.23: Two critical states during crack propagation with increasing  $\delta$ , redrawn from [29].

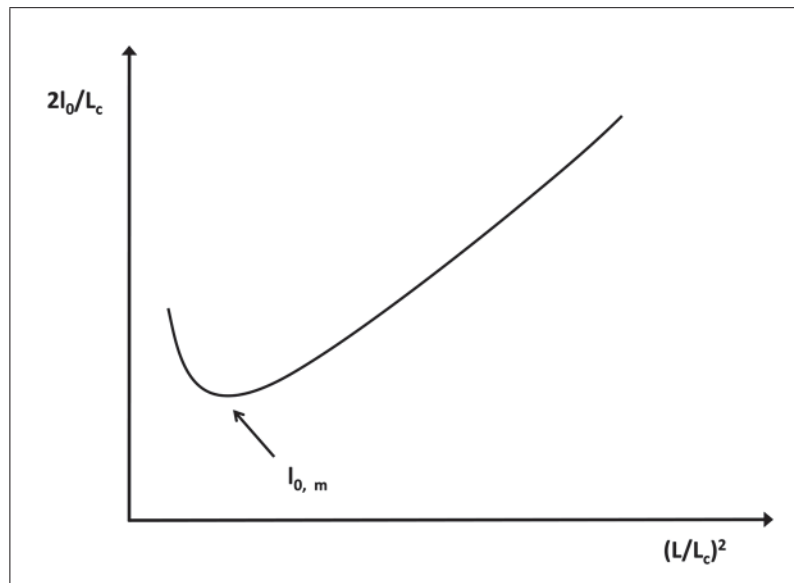
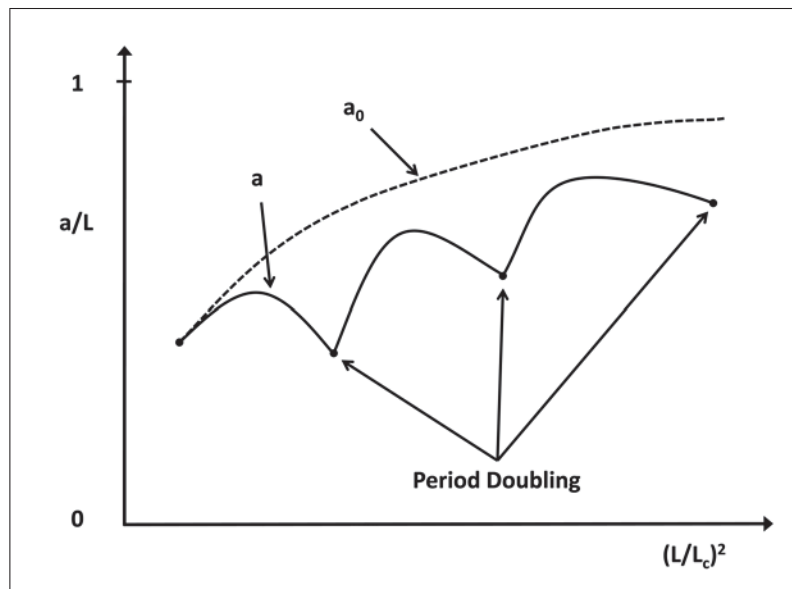


Figure 2.24: Determination of minimum spacing, redrawn from [31]. The expression  $\frac{L}{L_c}^2$  is proportional to time. After the cracks start to initiate the optimum spacing decreases. The global minimum determines the optimum spacing observed in the material.

system. For this purpose equations of mechanical equilibrium were solved by means of the FE method in order to determine  $U$  for each combination of  $l$  and  $a$ . Afterwards, the global minimum of the potential energy was identified and the corresponding values of  $l$

and  $a$  were designated  $l_0$  and  $a_0$ . This analysis was carried out for various stages of growth. The results found by Jenkins and depicted in Figure 2.24 show that at early stages of growth no sufficient stress level has developed to cause cracking. Shortly thereafter, however, cracks initiate with a relatively large optimum spacing. This spacing decreases with time which is equivalent to the initiation of additional cracks. After reaching a minimum spacing  $l_{0,m}$  the system would prefer larger spacings. These are not achievable, however, because cracks already initiated cannot disappear. For this reason, the energy of the system will be at higher levels than the optimum states determined for later stages of growth. The spacing  $l_{0,m}$  is therefore the one which will be observed at the surface.  $L$  refers to the thickness of the layer. In Figure 2.24  $l_0$  and  $L$  are scaled with  $L_c$ , which is the Griffith crack length for a shrinking solid and a measure for the depth of diffusion of heat necessary to initiate cracking. The term  $(\frac{L}{L_c})^2$  is proportional to time.



**Figure 2.25:** Period doubling during crack propagation in a growing layer, redrawn from [31]. The black line is valid for a crack array with the optimum initial spacing which is illustrated in Figure 2.24. The dashed line corresponds to crack configurations with the current optimum spacing in each time increment.

Based on this knowledge crack growth itself was investigated, starting with equal cracks in a distance of  $2l_{0,m}$  from each other and a length of  $a_{0,m}$ . The configurations with  $(l_{0,m}, a)$  correspond to higher potential energies than the configurations with  $(l_0, a_0)$ , which would cause the lowest potential energy possible at the specific time but have already been ruled out by the growth and initiation history of the system. Thus, instabilities in the growth regime are possible. In accordance with the findings of Nemat-Nasser, the cracks in the

array will propagate and maintain equal length until a criterion for instability is fulfilled. Jenkins found that the crack array in the solidifying layer undergoes several instabilities as shown in Figure 2.25, each accompanied by a period doubling, viz. every second crack stops at each instability. In Figure 2.25 the crack length  $a$  is scaled with the current thickness of the layer. The dashed line corresponds to  $a_0$ , the crack length at a specific time and the optimum spacing  $l_0$  at this time.

The idea of an optimum initial spacing is significant for the present study. An approach based on the potential energy was tested but not further pursued since energy approaches are too difficult to implement in case of fatigue. This is because the energy changes during each loading cycle and no concepts which consider this issue have been developed so far. Concepts based on the stress intensity approach to LEFM are more advanced and have been used for this work, as will be discussed in Chapter 3.

# Chapter 3

## Modeling

The FE models designed in the course of this thesis were implemented with the commercial simulation package Abaqus. Furthermore, the FE package Zmat [32] was utilized to model the material behavior, Matlab and Mathematica were used for data fits and the programming languages Java and Python have been crucial for the automated design of the modeling path.

The simulation of crack arrays is rather complex and was split into issues that can be treated individually. Beside the implementation of theoretical knowledge on crack arrays, focus was laid on parameterization, automation and performance to be able to conduct parametric studies and simulations of crack propagation on a reasonable time scale. Figure 3.1 shows a flow chart with the individual units and their correlation. The single units will be explained in detail in the following sections.

### 3.1 Preliminary Studies: T-Model and S-Model

#### 3.1.1 Model Design

Because of performance issues, the computation of temperature distributions, residual stresses and crack propagations were separated into three distinct models. The Temperature-Model (T-Model) and the Stress-Model (S-Model) will be described at this point. Since the temperature distribution in even die areas is one dimensional and thus the stress distribution as well, both models consist of a single row of elements which allow the computation of temperature and stress distributions from the surface to a depth of 50 mm. For the T-Model DC2D4 elements are used, i.e. 4-node linear elements typical

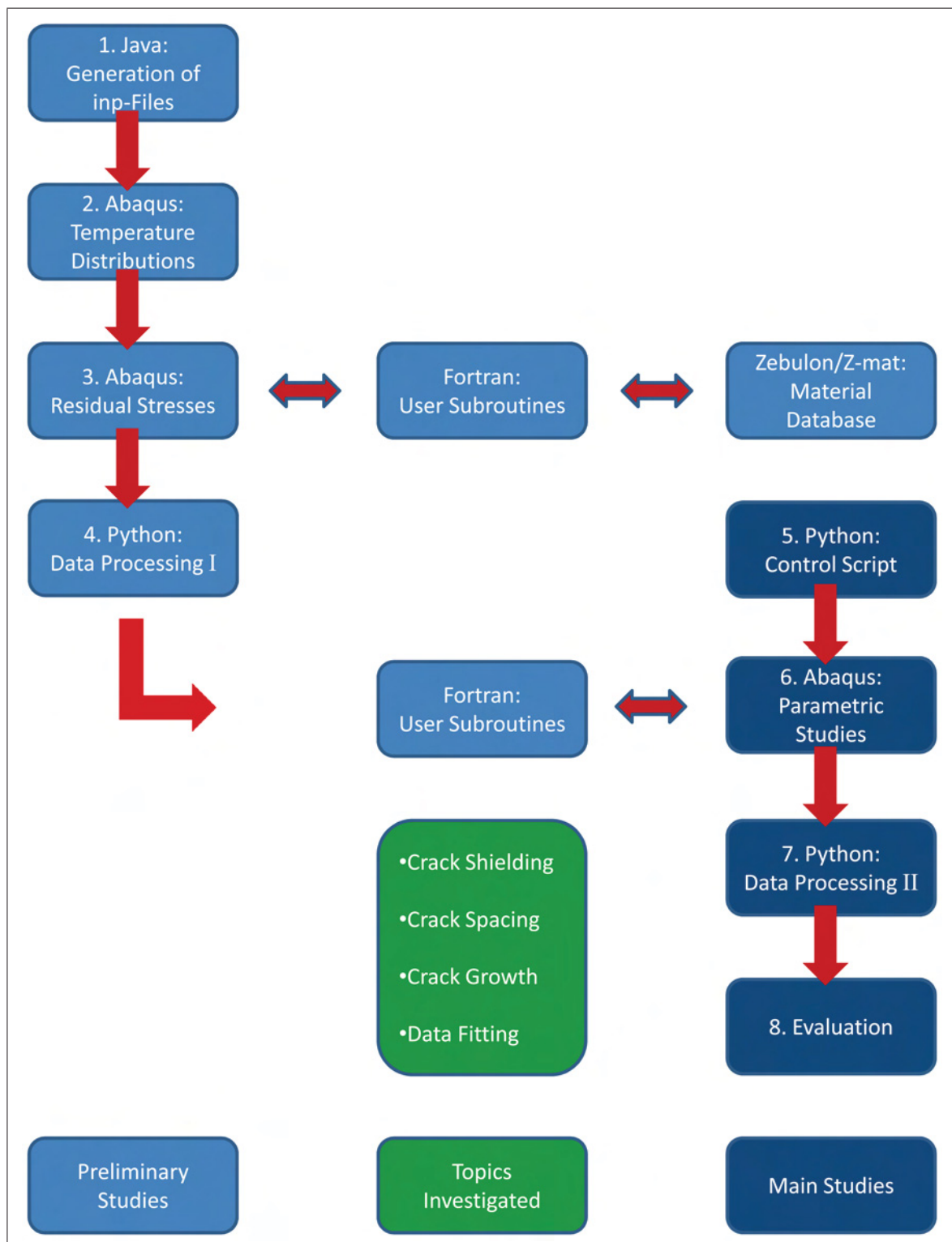


Figure 3.1: Flow chart of the modeling path.

for heat transfer analyses. For the S-Model quadratic plane strain elements with reduced integration are used (CPE8R). The mesh size at the top is 0.1 mm and the areas below a depth of 10 mm are meshed with a seed size of 0.3 mm whereas the model width is kept constant at 0.1 mm. Since data on temperature distributions computed by the T-Model are transferred to the S-Model the same mesh should be applied to both models.

The T-Model is used to simulate heat transfer from the melt in the solidification step to the die, thermal fluxes within the die and heat transfer to the surroundings after the ingot has been removed. This is done in multiple steps by applying appropriate boundary conditions. The S-Model is used to compute the material response to thermal loading. For this purpose boundary conditions were set to block deformation at the sides and the bottom of the model. In this simple model this has the same effect as periodic boundary conditions allowing to determine stresses and residual stresses.

### 3.1.2 Generation of Input Files

The FE solver used to compute the FE models receives Input-Files (inp-files). Inp-files are text files consisting of a list of keywords each followed by specific data lines. The solver reads inp-files and processes their commands. Usually the model is designed with a graphical user interface (GUI) provided by Abaqus CAE which creates the corresponding inp-file of the model and passes it on to the solver. It is possible, however, to bypass the GUI and write inp-files manually as it has been done for the creation of T-Models and S-Models.

A single casting cycle consists of 10 different steps, which are described in detail in [4]. Every additional casting cycle is equivalent to another set of these steps. Thus, an inp-file was prepared for the first casting cycle both for the T-Model and the S-Model. To be able to generate inp-files for an arbitrary amount of cycles and to adjust these files to current requirements, a package of Java programs was written and used for the preliminary studies, viz. the computation of temperature and stress distributions for given sets of parameter variations. The parameters investigated are hardness, thermal conductivity ( $\lambda$ ) and thermal expansion ( $\alpha$ ).

The package was used to generate a root file for each model and to create several sets of inp-files based on the root files afterwards.

In this study the temperature and residual stresses were computed for 100 casting cycles. After 100 cycles one can safely assume that all state variables will no longer change, i. e. a stable cycle has been reached. The corresponding inp-files consist of 1000 steps and about 30000 lines. The names of the inp-files with varying parameters were stored in a

**Table 3.1:** Important thermo-physical material parameters of a X38CrMoV5-1 [7].

$Temperature/[^{\circ}C]$	20	100	200	300	400	500	600	700
$E/[GPa]$	210	199	194.5	190	188	183	183	183
$\lambda/[W/mK]$	24.5	26.0	27.7	28.9	29.5	29.5	29.1	29.2
$\alpha/[(1/K)10^{-6}]$	10.5	11.13	11.75	12.28	12.73	13.13	13.33	13.58

batch file and started sequentially with Abaqus.

There is an alternative concept: Abaqus provides the possibility to conduct parametric studies based on parameterized inp-files in combination with a Python script that manages the parametric study. This script contains predefined functions especially developed for parametric studies. For more information, the reader is referred to the Abaqus documentation. This concept, however, does not consider the variable number of load cycles which necessitates an additional program that generates the parametric root file before the actual parametric study begins. Both concepts offer similar functionality but in the long term the built-in concepts of Abaqus, based on Python, should be preferred.

### 3.1.3 Temperature Distributions

The die casting process itself was investigated extensively in [4] and [7]. In the course of this thesis the models developed by Ecker and Leindl were adapted and modified leading to the T-Model and the S-Model. As mentioned above, the T-Model has been reduced to a one dimensional heat conduction model which is an appropriate means to handle the temperature conduction problem in the dies without spending more resources than necessary.

The T-Model was used to compute temperature distributions for nine different values of thermal conductivity<sup>1</sup>. Table 3.1 shows the thermo physical properties of the investigated material. The other material parameters used in this study result from variations of these values as depicted in Table 3.2.

### 3.1.4 Residual Stresses

The S-Model is coupled with the material modeling data base Zmat which is a standard feature of the FE package Zebulon. This coupling is done via the User Subroutine UMAT,

---

<sup>1</sup>In [4, 7] it was found that  $\alpha$  does not have an influence on the transient temperature field.



**Table 3.2:** Variations of  $\lambda$ ,  $\alpha$  and hardness used for the studies in the present work.

$\lambda$	$\alpha$	HARDNESS
-50%	-50%	44HRC
-25%	-25%	47HRC
-15%	-15%	50HRC
-5%	-5%	
+0%	+0%	
+5%	+5%	
+15%	+15%	
+25%	+25%	
+50%	+50%	

which itself enables the simulation of material behavior. Nevertheless, Zmat offers superior functionality. This way the material model developed by Ecker [4] is implemented into the simulation evaluating the plastic response of the material parallel to the stresses computed by Abaqus. The complex elasto-viscoplastic material model consumes significant hardware resources making this model the most time consuming one in the whole chain of simulations. This is the reason why the simulations in the course of this work were done with three different models, i.e. the T-Model, the S-Model and the Crack Propagation Model (C-Model). The first two models could be implemented as one dimensional models allowing the simulations to finish in reasonable time. One computation with the S-Model takes only 2.5 hours, whereas this model combined with the C-Model would lead to computation times in the range of days or weeks. Because multiple C-models are necessary to simulate the evolution of crack arrays, computation time is a crucial issue.

Since mathematical relations between material parameters and e.g. the service life time are to be found as the final aim of this work, studies are carried out on every combination of hardness,  $\lambda$  and  $\alpha$ . The data shown in Table 3.2 represent the parameter grid this work is based upon which results in 243 parameter combinations. Thus, a total of 243  $\sigma_{res}$  distributions was computed and this way, enough data points were available for xyz-data fits, which provide a means for quick estimations of the influence of these parameters. If no mathematical fit functions can be found the distribution of the data points itself will provide valuable information.

### 3.1.5 Data Processing I

The preliminary studies give information on temperature and  $\sigma_{res}$  distributions for different material parameters. This information has to be transferred to the C-Model. The

easiest way to do this is to take advantage of the User Subroutines UTEMP and SIGINI which allow to impose temperature and stress distributions on the nodes of a model as initial conditions before the actual computation. Since the temperature distributions in contrast to  $\sigma_{res}$  extend over the whole model and because they can be approximated by polynomials quite easily, the obtained data is fitted with Mathematica. In a first step a Python script is used that reads data from the output data bases (odbs) which contain the computed results of the FE analyses and writes those data systematically to text files. These files are read by Mathematica which determines the coefficients of the fit polynomial. The polynomials are written to User Subroutines generated from a template file. In case of the stress distributions a Python script is used to read the data and to write them to a template directly. Data fitting is not possible because only complex functions or spline curves would be able to fit the shapes of the distributions. Piecewise linear approximations are appropriate means to transfer the distributions computed by Abaqus at the integration points to the subroutines, especially if reasonably fine meshes are used in the models.

Additionally, the envelopes of the stress distributions of the last solidification step are needed for parallel computation of  $\Delta K_{eff}$  in the C-Model<sup>2</sup>. The envelope of the distributions of all increments yields the minimum stress values with respect to depth. This is also taken care of by a Python script that reads data, computes the envelopes and generates a module that can be loaded by the control script (see the schematic in Figure 3.2).

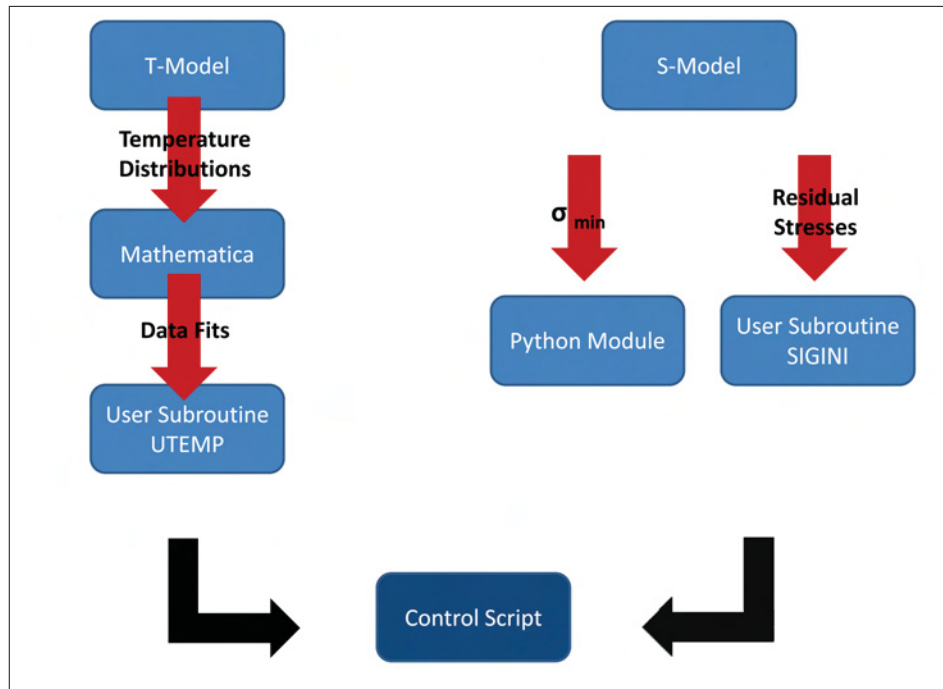
## 3.2 The Architecture of Abaqus

The investigations on crack arrays are facilitated in this thesis by the scripting language Python and the scripting interface built in Abaqus CAE. Basic knowledge of Python and the architecture of Abaqus are preferable in order to follow the modeling path and the discussions in Chapter 5. Some basic information is provided in this section. Further information can be obtained from the official Python documentation [33], the Abaqus Scripting User's Manual [34] and the Abaqus Scripting Reference Manual [35].

An FE analysis with the commercial package Abaqus passes through three distinct steps illustrated in Figure 3.3. The first one is the preprocessing, where the geometry, loads, boundary conditions, mesh, etc. of the model are defined. This is usually performed using

---

<sup>2</sup>As will be discussed in Section 3.3.1  $\sigma_{min}$  is dependent on step time and depth.



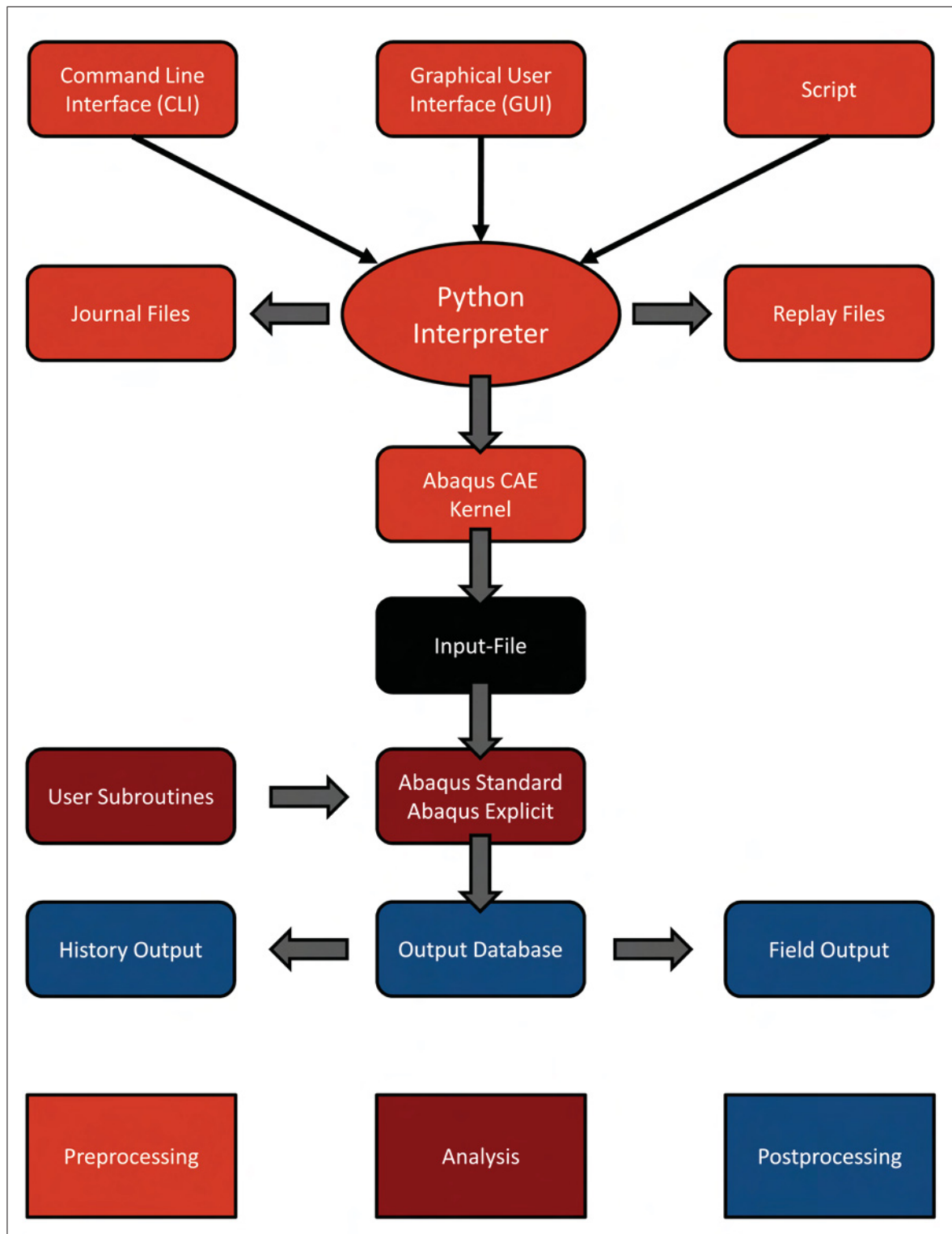
**Figure 3.2:** Overview of data transfer to the main script.

the graphical user interface (GUI), provided by Abaqus CAE, the part of the package developed specifically for the preprocessing. Abaqus CAE was not available till the mid-1990s so, as was common for FE programs of that time, the model was built just by manually writing a text file containing lines with distinct keywords, each followed by associated data lines. Today, the preprocessor does nothing else than generating this inp-file by translating the GUI input into a text file. This inp-file is the link between Abaqus CAE and the solvers Abaqus Standard and Abaqus Explicit, needed for the next step, viz. the analysis of the model. The solver is chosen by the user according to the type of the problem, e.g. static or dynamic. Abaqus CAE then passes the generated inp-file to the chosen solver. After analysis has completed the results can be visualized and postprocessed in Abaqus Viewer, which has been integrated into Abaqus CAE.

As depicted in Figure 3.3 the entire preprocessing via Abaqus CAE communicates with the Python interpreter. Python is an object oriented programming language Abaqus CAE was developed with. GUI inputs are internally translated to Abaqus-Python commands and these are processed by the interpreter. Python is a language using bytecode which is generated during interpretation and in case of Abaqus passed to the kernel, the brain of Abaqus. In other bytecode based languages like Java, a compiler generates this code as a part independent of the source code. In Python the bytecode is just kept in the main

memory and it is thus dependent on the interpreter. The kernel processes the bytecode and operates on the bytecode based model data bases (mdb) and output data bases (odb). The recorded Python commands used in a model or a session, respectively, are stored in the journal and replay files.

A very useful feature of Abaqus CAE, used extensively in this study, is the possibility to access the Python interpreter and thus the kernel by using the scripting user interface. This way models can be built directly by means of Abaqus Python commands. This procedure entails far greater efforts but has a crucial advantage over GUI modeling. The user can take advantage of the full capabilities of Abaqus CAE as well as the high-level programming language Python. Parameterized modeling and automated procedures involving model data bases (mdb) and output data bases (odb) are hence possible.

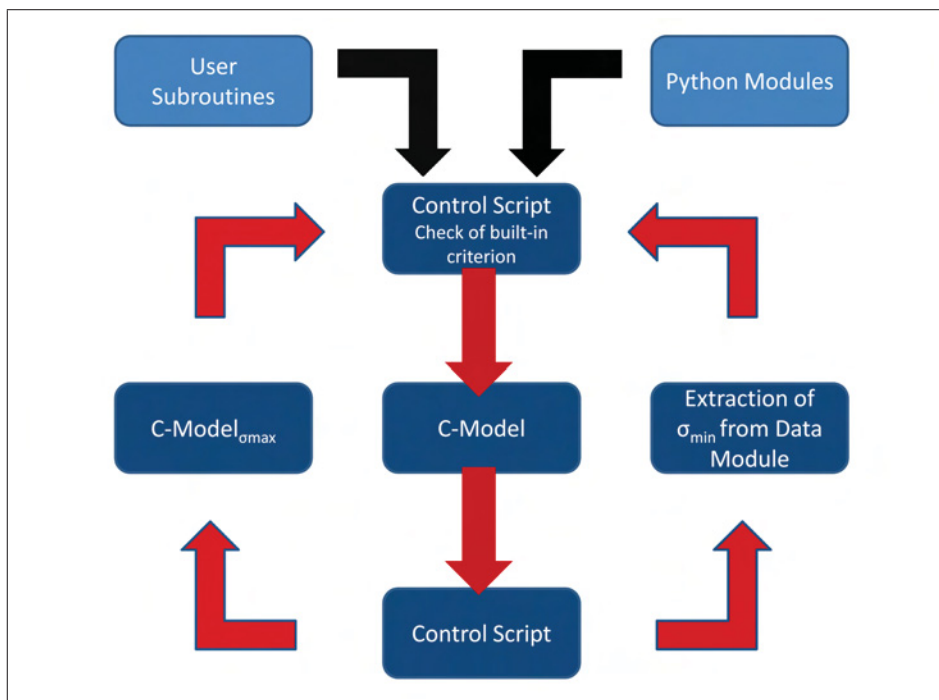


**Figure 3.3:** Abaqus schematic, modified from [34]. The orange part illustrates the design of the Abaqus preprocessor (Abaqus CAE). The red section describes the solvers provided by the Abaqus FE package. The blue section refers to postprocessor which is integrated in Abaqus CAE and the two types of output, i.e. field and history output.

### 3.3 Main Studies: Control Script and C-Model

The C-Model was used to simulate propagation of cracks in crack arrays. This model corresponds to an even section within the die and uses periodic boundary conditions which permit the reduction of the model to a size that is still representative for the entire array. A model with periodic boundary conditions can be thought to be continued infinitely. This means that although not all cracks in the die are included in the model, their influences are still considered in the simulation.

Because of the chosen modeling design, the simulation of crack propagation consists of a chain of individual computations that provide data used by the subsequent one. Thus, the generation of a single C-Model is not sufficient. This issue is handled by the Control Script (see appendix), a program written in Python, that creates C-Models in loops, evaluates generated data and creates C-Models based on built-in criteria and predefined settings. Before the Control Script builds the first C-Model, user defined modules are loaded containing the data discussed above. Figure 3.4 shows the functionality of the Control Script.



**Figure 3.4:** Relation between Control Script and C-Model. When the script is started user defined modules are loaded. These contain data required for the application of a built-in criterion and functions needed during the execution of the script.

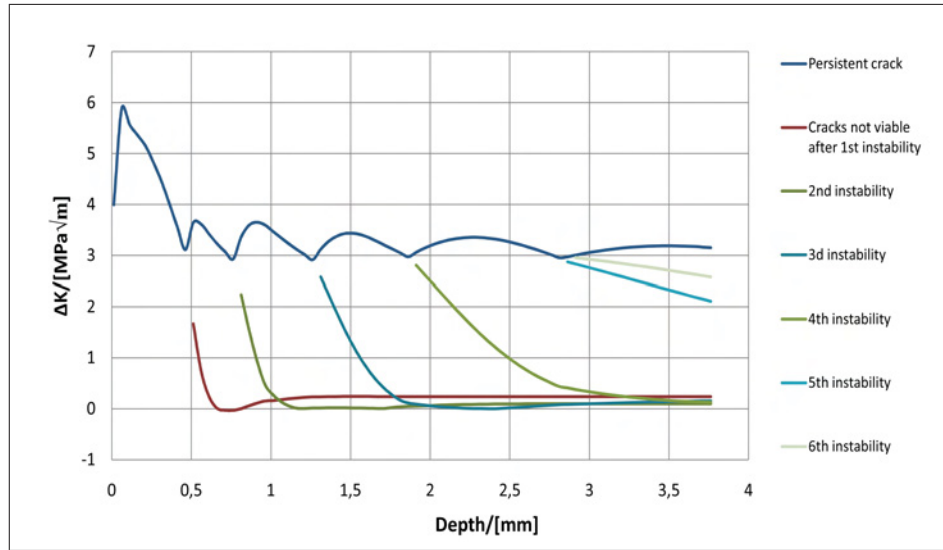
During the application of the Control Script it has to be checked for each crack config-

uration whether the array would propagate. For this purpose the concepts discussed in Section 2.8 were reviewed in terms of fatigue. Although derivations from Section 2.8 do not apply in this case, the views on energy and the stress field persist. As discussed for monotonic loading, the system will always choose the path to the lowest possible energy state. For monotonic loading like in rock it was debated what happens if stable crack growth satisfies an equilibrium condition [29]. It was concluded that crack growth can either stay stable, change to instable propagation or become critical, which causes other cracks to stop. This debate was conducted in terms of  $K_c$  and  $G_c$  which mark equivalent equilibrium conditions. The corresponding quantity in fatigue is  $\Delta K_{th}$ . This quantity, however, marks the borderline between no crack propagation and stable crack growth. Thus, the problem is not equivalent but still it can be treated in a similar way which gives rise to even simpler conclusions.  $\Delta K_{eff}$  is connected to dislocation gliding at the crack tip. When gliding stops, ductile crack propagation, as in the case of the material investigated, is no longer possible. Unstable propagation is not an issue since  $\Delta K_{eff}$  is far below  $K_c$ . This means that the system cannot lower its potential energy any more since the array as a whole is not viable. On the assumption that all cracks initiate at the same time with a homogeneous spacing, every second crack will stop at equilibrium  $\Delta K_{eff} = \Delta K_{th}$  if  $\frac{\partial \Delta K_{eff}}{\partial a} < 0$  which is always fulfilled in the case of equilibrium in casting dies<sup>3</sup>. There is no configuration leading to lower potential energy. The continuing crack in a unit cell will experience a jump of  $\Delta K_{eff}$ , whereas the other one will be shielded immediately and drop significantly below  $\Delta K_{eff}$ . The case that all cracks stop does not occur until the system cannot lower its energy any more by crack propagation. This is assumed to mark the final configuration of the array. Figure 3.5 shows a result based on this concept. The points where the configuration of the array changes, viz. where cracks stop growing are designated as instabilities. This expression refers to the crack configuration only and does not imply instable crack growth. This nomenclature will be kept in the following sections.

The stress intensity factor is continuously compared to the threshold value. If  $\Delta K_{eff} > \Delta K_{th}$  the array is stable and the configuration will not change. In order to be able to use the criterion it is necessary to know the stress ratio at the crack tip of the propagating crack in the unit cell as well as the relation between  $K_{max}$ , which is computed by Abaqus, and  $\Delta K_{eff}$ .

---

<sup>3</sup>At first  $\frac{\partial \Delta K_{eff}}{\partial a}$  is positive but the first equilibrium is satisfied after a peak and thus at negative values of the derivative.



**Figure 3.5:** Progression of  $K$  for all cracks in an array with an optimum minimum initial spacing of  $230 \mu\text{m}$ . The crack configurations where cracks stop are designated as instabilities. The stress intensities of cracks which have stopped drop significantly upon the propagation of the other cracks.

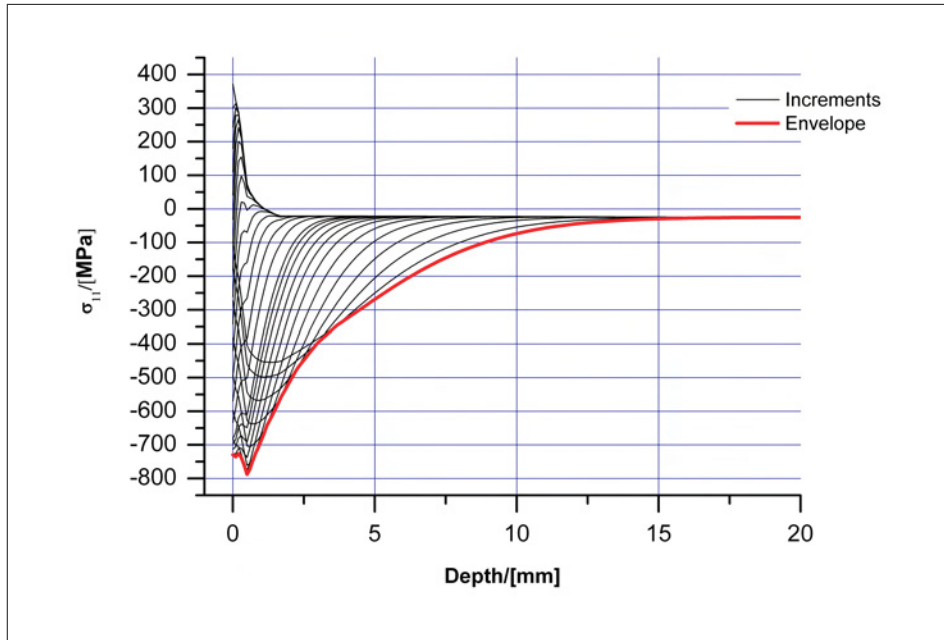
### 3.3.1 Determination of the Stress Ratio

#### Minimum Stresses

$\sigma_{min}$  occurs during solidification of the melt in each casting cycle. Due to the considerable difference in temperature of melt and die the temperature distribution and hence the stresses change significantly in each increment computed by Abaqus. Thus, the values of  $\sigma_{min}$  of different depths in the die arise at different times. The distribution of  $\sigma_{min}$  with depth can be found, however, by designing the envelope of all increments. Figure 3.6 depicts the stress distributions after each individual increment of the FE analysis. The envelope which is equal to the distribution of  $\sigma_{min}$  is marked red. After the first increments there are still tensile stresses at the surface since residual stresses shift the distribution in that area to positive values. The peak decreases until a minimum value is reached. From this point on the stresses become less negative because with time heat flow partially evens out the temperature differences, i.e the temperature distribution reaches deeper into the die while consuming thermal energy stored in the surface region to some extent. The spike near the surface is due to the residual stresses which add to the stresses caused by thermal expansion.

Because the  $\sigma_{min}$  distributions are entirely in the pressure regime all cracks are supposed





**Figure 3.6:** Envelope of all increments of a solidification step. This envelope corresponds to the distribution of  $\sigma_{min}$

to be closed and it is assumed that they do not change the distributions<sup>4</sup>. Therefore the envelopes are determined from the S-Model directly which does not contain any cracks at all. The evaluation is performed by means of the same script the subroutines were generated with and no extra programming step is necessary. The envelopes are approximated by piecewise linear fits and written to a text file. This data module is a simple form of a data base, consisting of if-conditions specifying the data sets and the sections of the fit functions.

As mentioned above, this module can be loaded and used in the control script whenever necessary. After a crack array has been computed, the database is called and  $\sigma_{min}$  at the crack tip of the propagating crack is loaded. In order to obtain the stress ratio,  $\sigma_{max}$  must be computed at the same position.

---

<sup>4</sup>Due to plastification the cracks are open when unloaded and pressure is necessary to close them. This changes  $\sigma_{min}$  slightly.

## Maximum Stresses

The maximum stresses occur in the spraying step of each casting cycle. There are three spikes during this step since the spray nozzle passes each point of the surface three times. Before the first cycle is started the die is heated to about 120°C to 250°C and is kept at that value in the area of the cooling channels within the die. In this study  $T_0$  is 120°C which matches the studies carried out in [7, 4]. The last pass of the nozzle gives rise to the largest tensile stresses in the mold. The temperature distribution and thus the stress distribution after the final pass are used in the C-Model. The  $\sigma_{max}$  distributions start in the tensile regime, which causes the cracks to open. Therefore, the influence of cracks on the stress field is essential and  $\sigma_{max}$  must be computed with a model related to the C-Model. In this case, the propagating crack must be kept closed, however, since  $\sigma_{max}$  in the method chosen is by definition the stress of the uncracked model at the same position as the crack tip. All the cracks in the surrounding area and their impact on the stress field have to be taken into account.

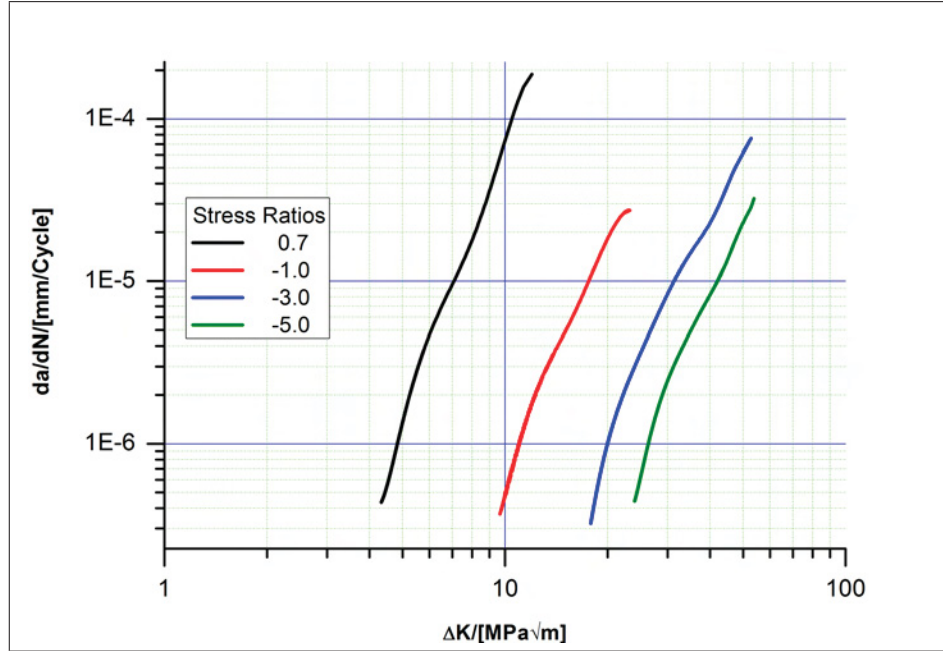
This gives rise to the design of a C-Model differing from the standard one, designated as  $C - Model_{\sigma_{max}}$ . Due to the periodic boundary conditions the absence of a crack affects the whole crack array. For this reason, the unit cell must be enlarged to make sure the closed cracks do not influence each other. In case of an initial unit cell, consisting of one crack only, the cell will be enlarged by a factor of 10 accommodating 9 identical cracks. A configuration with 9 cracks is chosen because that way  $\Delta K$  of the cracks farthest apart from the closed crack are at the level of the cracks in the corresponding C-Model. This procedure guarantees that the closed cracks in the array described by the  $C - Model_{\sigma_{max}}$  do not influence each other and that  $\sigma_{max}$  is valid. The geometry of crack number 1 will be modeled but no crack seam will be defined. Thus, the geometry affects the mesh but no crack is present at this position in the simulation. The node at the crack tip will be stored in a set as it is usually done in the C-Model, which enables the Control Script to access the required stress data in the output data base of the current  $C - Model_{\sigma_{max}}$  easily.

Distributions of  $\sigma_{min}$ ,  $\sigma_{max}$  and R will be discussed in Section 4.3.

### 3.3.2 Determination of the Effective Stress Intensity Range

In order check the criterion for the stability of the forming crack array it is necessary to determine a relation between  $K_{max}$  computed by Abaqus and  $\Delta K_{eff}$  which must be compared to a threshold value obtained on the basis of experimental data. In the course of finding this relation the data obtained by Leindl [7] who determined crack propagation

curves for different stress ratios is used. The data is shown in figure 3.7. The horizontal



**Figure 3.7:** Crack propagation curves for a X38CrMoV5-1 hot work tool steel with a hardness of 44 HRC for various stress ratios [7].

shift of the curves with  $R$  indicates the presence of crack closure as discussed in Section 2.7. Due to crack closure the crack tip is subjected to loads less than predicted from external loading conditions. The impact of crack closure depends on the stress ratio and therefore its determination is necessary as mentioned above. If the effective dynamic stress intensity factors at the crack tips are known and a new chart is printed in terms of  $\Delta K_{eff}$  the curves are supposed to be on top of each other. This procedure was investigated by Huang and Moan [8].

The modus operandi in this study was to calculate a  $\Delta K_{eff}$  curve from  $\Delta K$  data for a stress ratio of 0.7 by application of an adapted version of the relation found by Elber and implemented in [7]. This correction is assumed to give precise values in a stress ratio range from 1.0 to -1.0. The data corresponding to the other stress ratios were in a first step transformed to  $K_{max}$  curves by application of Equation 2.24. In a second step the resulting curves were shifted individually to get a set of superimposed stress propagation curves as shown in Figure 3.8. The focus when shifting the curves was led on the lower portion of the curves which is in the present case predominant considering life time. The upper parts of the curves usually differ from each other to some extent.

The coefficients applied for the curve shift, correspond to the ratios  $\frac{\Delta K_{eff}}{K_{max}}$  specific for each

**Table 3.3:** Polynomials used for the data fit required in order to relate  $K_{max}$  with  $\Delta K_{eff}$ . The polynomials are depicted in Figure 3.9.

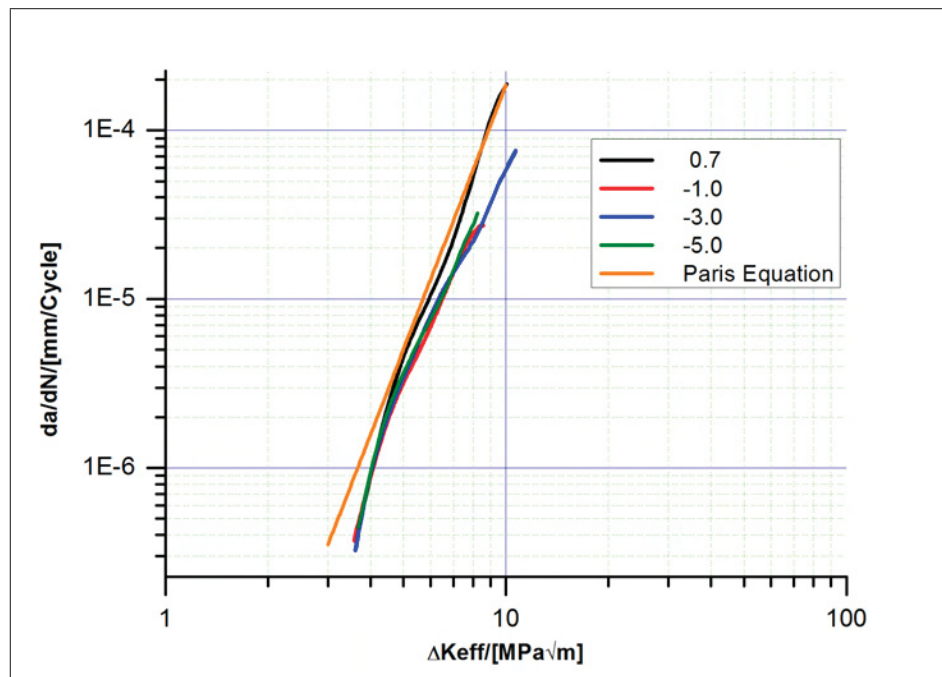
RANGE OF STRESS RATIO	POLYNOMIAL
0.7 to -1.0	$0.543247 - 0.325812 * R - 0.129059 * R^2$
-1.0 to -3	$0.705 - 0.035 * R$
-3.0 to -5	$0.645 - 0.055 * R$
-5.0 to -6.455	$0.645 - 0.055 * R$
lower than -6.455	$\frac{\Delta K_{eff}}{K_{max}} = 1.0$

stress ratio. For  $R = 0.7$  the ratio was calculated directly by Equation 2.24. These ratios were used to find a general relation between  $\frac{\Delta K_{eff}}{K_{max}}$  and the stress ratio for a X38CrMoV5-1 hot work tool steel. The results are depicted in Figure 3.9.

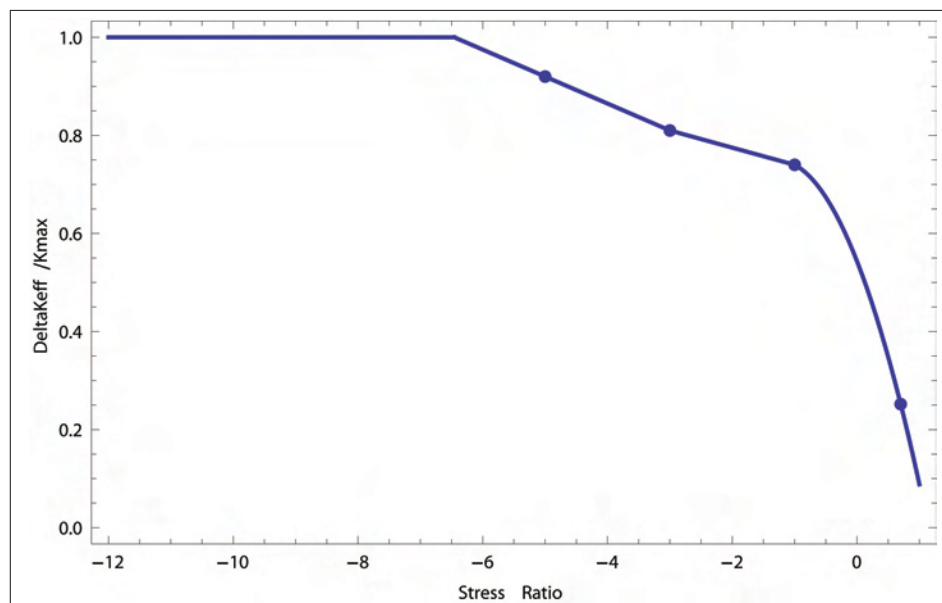
Since no satisfying curve fit can be found a curve is defined consisting of the polynomials listed in 3.3. The curve is likely to approach 1 asymptotically at very negative stress ratios. Because no data is available for stress ratios below -5, however, a conservative curve progression is chosen. Below -5 the curve is linearly propagated by the preceding 1st order polynomial. The value of this function reaches 1.0 at a stress ratio of -6.455. At stress ratios lower than -6.455  $\frac{\Delta K_{eff}}{K_{max}}$  is supposed to be 1.

The piecewise fit with polynomials is regarded as the best solution in this case because no other method would provide a better fit for the present data. This is due to the deviant data point at  $R = -1$  which would decrease the quality of a fit with a single function. Even in case of a weighed fit function that puts more weight to the other data points the results would not be better. The deviant data point originates from experimental data and its deviation must be attributed to a physical influence. Thus, the most appropriate approach is to use linear fits in the discussed data range.

Figure 3.8 also shows the Paris relation determined by Leindl in [7]. The procedure for shifting the curves was different and other corrections beside the modified Elber correction were applied. Thus, the Paris relation deviates to a minor extend from the  $\Delta K_{eff}$  curve determined in this thesis. Nevertheless, the coefficients C and m of the Paris equation determined in [7] are used for this work, since it is not known which method is more precise and Leindl has already obtained reasonable results based on his data. Furthermore, the deviation from the present curve is small and in case the  $\Delta K_{eff}$  curve determined in the present work is more precise the results for the studies on service life are conservative which would not be true for the opposite case when C and m would be determined anew. Crack propagation in ductile materials is supposed to be based on cyclic plastic deformation at the crack tip. Thus,  $\Delta K_{th}$  is connected to the initiation of cracks due to cyclic plasticity, which in turn is dependent on the emission and gliding of dislocations



**Figure 3.8:** The crack propagation curves are shifted in order to obtain a  $\Delta K_{eff}$  curve which is compared to Paris relation determined in [7].



**Figure 3.9:** Data obtained from the shift of crack propagation curves (Figure 3.8) is used to define  $\Delta K_{eff}/K_{max}$  as a function of R.

at the crack tip. The emission of dislocations depends on the shear modulus, which is almost the same for different hardnesses. For this reason,  $\Delta K_{th}$  can be assumed to be the same for X38CrMoV5-1 hot work tool steels with different hardnesses. In general, it was found that even different types of steel have nearly the same threshold value of  $3.0 \text{ MPa}\sqrt{m}$  [36]. This value is used in this study. The curves in Figure 3.8 consist of data points greater than  $\Delta K_{eff}$ . The reason why the threshold region is not represented is the significant scatter of data points. Furthermore, it should be considered that in the range of  $3.0$  to  $3.5 \text{ MPa}\sqrt{m}$  a corrected form of the Paris equation (Equation 2.33) should be applied, at least in cases where this region is significant for fatigue life estimations. Anyway, if the correction is not applied the estimates are conservative.

It was shown by Leindl [7] that the plastic zone size correction discussed in Section 2.1.2 does not have significant influence on  $\Delta K_{eff}$ . Thus, this correction was not implemented in the modeling path.

### 3.3.3 The Relation between Crack Propagation and Load Cycles

As mentioned above, crack propagation is computed step by step by a self controlled Python script, which determines the evolution of crack arrays. During analysis R and  $\Delta K_{eff}$  are computed parallel to crack propagation. This analysis was done for various data sets, each consisting of a unique combination of material parameters.

In addition to the output already mentioned, it is of scientific interest to know crack propagation rates in each step and the number of loading cycles corresponding to crack propagation between the C-Models. The information, at which stages of the formation of the arrays most of the fatigue life is spent, is a major issue in this study.

The design of the chosen modeling path favors this kind of computations, where data are extracted and used for parallel evaluation of criteria or specific characteristics. The propagation itself is given by a fixed increment but the distribution of fatigue life is supposed to be almost independent from this incremental modeling, provided that the increment size provides an appropriate resolution of crack growth. The crack propagation rates for each C-Model can be determined easily from Equation 2.25.  $\Delta K_{eff}$  is computed automatically and the constants C and m are used as determined by Leindl [7]. The determination of load cycles corresponding crack propagation in the specific configurations of the arrays is a more complex task. In Section 2.3.1 an approach was introduced linking crack extension to load cycles based on the integration of Paris equation. In this case  $\Delta K$  was determined by  $\Delta\sigma$  which is the difference between the minimum and maximum remote load. The origins of the loads in the C-Models are thermally induced stresses and residual stresses.

For this reason, no remote loading exists and an alternative approach has to be chosen, since Equation 2.27 is not applicable.

On the one hand crack propagation curves are available for the application of Paris equation, on the other hand the progression of  $\Delta K_{eff}$  with increasing crack length is recorded in the course of the FE analyses. With this data available it is possible to integrate Paris equation for regime II, either as a whole after a simulation in a separate step or as a kind of “live evaluation” during simulation. The latter method was judged to be the better one because this way the number of loading cycles will be computed automatically for each increment of propagation based on piecewise linear fitting of the distribution of  $\Delta K_{eff}$  (Equation 3.2). The schematic in Figure 3.10 is supposed to give an impression of this procedure. Due to the small increments of 0.05 mm these fits are more precise than a fit of the whole distribution. Furthermore, it is computationally less expensive to integrate this evaluation into the process instead of having to access all data in a separate postprocessing.

From Equation 2.25 the following expression can be obtained.

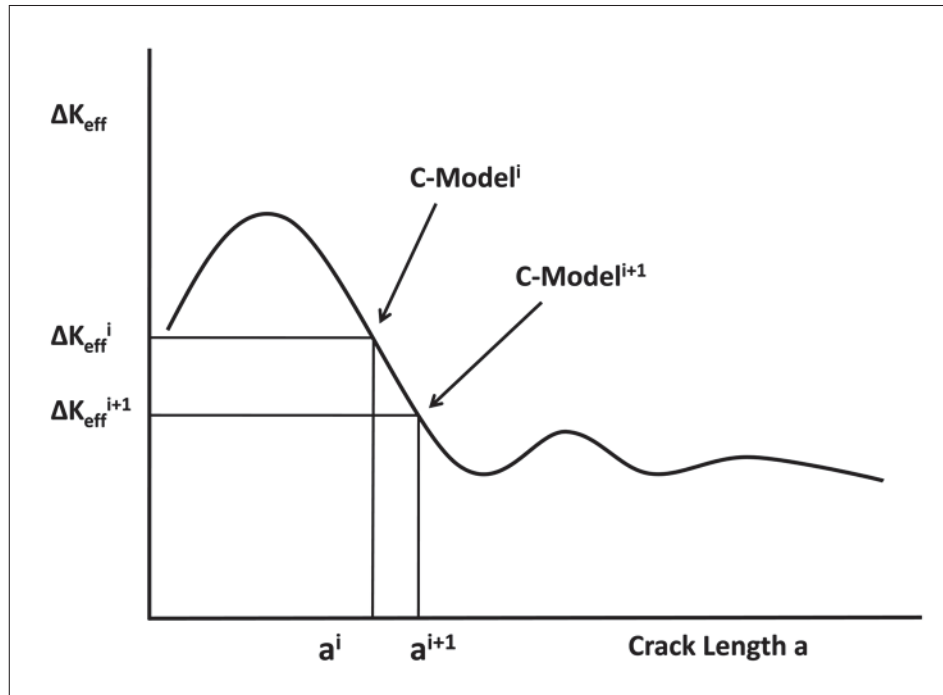
$$N_{End} - N_{Start} = \int_{a_{Start}}^{a_{End}} \frac{1}{C(\Delta K_{eff})^m} da \quad (3.1)$$

In Section 2.3.1  $\Delta K_{eff}$  is replaced by a term containing the cyclic remote loading, crack length and a geometry factor. It has already been pointed out that in the present work this term cannot be used, since the stresses in the die are not related to remote loads. The stress intensity factors, however, are calculated by Abaqus and the curve progression of  $\Delta K_{eff}(a)$  can be interpolated according to Equation 3.2. The index  $i$  refers to the increment of crack propagation that is to be evaluated. Stress intensity factors and crack lengths thus are to be taken from the C-Models before and after crack propagation (see Figure 3.10). Equation 3.2 can be substituted in Equation 3.1 and integration gives the relation needed to compute the number of loading cycles in each increment.

$$\Delta K_{eff}(a) = \frac{\Delta K_{eff}^{i+1} - \Delta K_{eff}^i}{a^{i+1} - a^i} (a - a^i) + \Delta K_{eff}^i \quad \text{for } a^i \leq a \leq a^{i+1} \quad (3.2)$$

$$N^{i+1} - N^i = \int_{a^i}^{a^{i+1}} \frac{1}{C\left(\frac{\Delta K_{eff}^{i+1} - \Delta K_{eff}^i}{a^{i+1} - a^i} (a - a^i) + \Delta K_{eff}^i\right)^m} da \quad (3.3)$$

$$N^{i+1} - N^i = -\frac{(a^i - a^{i+1})((\Delta K_{eff}^i)^{1-m} - (\Delta K_{eff}^{i+1})^{1-m})}{C(\Delta K_{eff}^i - \Delta K_{eff}^{i+1})(1-m)} \quad (3.4)$$



**Figure 3.10:** Progression of  $\Delta K_{eff}$ . For the integration of the Paris equation the progression is approximated by linear sections.



### 3.3.4 Parametric Studies

The Control Script is able to generate a modeling chain based on initial settings determining the type of study. A chain consists of dependent C-Models which are evaluated automatically. The following list summarizes the functionality and output variables provided by this design:

- Distribution of  $K_{max}$ .
- Computation of R based on  $\sigma_{min}$  and  $\sigma_{max}$ .
- Distribution of  $\Delta K_{eff}$ .
- Integration of Paris equation.
- Criterion for crack propagation.
- Automated data processing and output.
- Automated simulation of crack arrays for a set of material parameters.

Since the entire functionality of a programming language is available the Control Script offers a high degree of flexibility. Therefore, it can be modified to generate various modeling chains fulfilling different tasks. In order to reach the aim of this study, i.e. to identify interconnections between material parameters and the evolution of crack arrays, a range of different studies is carried out, which will be discussed in Chapter 5.

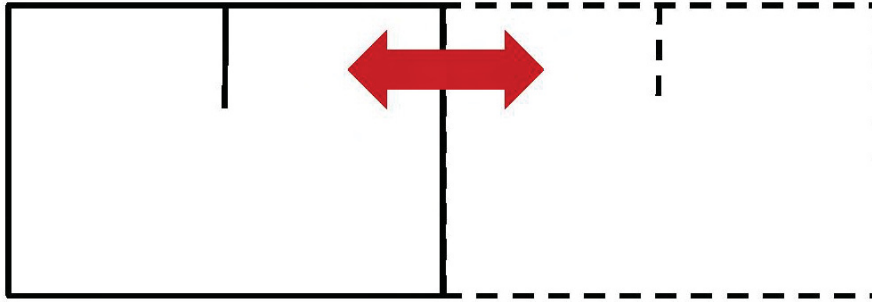
The computation time of the studies conducted with the Control Script is in the range of days. The data on service life of a set of 25 materials is usually obtained within 2-4 days, provided all necessary data like the distributions of temperature and residual stresses are available. The computation time with the S-Model for the corresponding residual stresses is about 2 days. The analysis of the T-Model takes about half an hour per temperature distribution.

### 3.3.5 Determination of the Characteristic Minimum Spacing

In order to investigate possible effects of variable initial spacings on service life, a method is devised which relates the spacing to the stresses in the die material acting as driving force represented by  $\Delta K_{eff}$ .

In this study the Control Script and C-Models are used in a specific mode. All C-Models in the study contain one 100  $\mu m$  long crack and stresses are impressed due to the results

of the S-Model. In the beginning the model width is set to  $50 \mu m$ , then the width is increased until  $\Delta K_{eff}$  reaches the threshold value  $\Delta K_{th}$ . Because of the periodic boundary conditions the model width corresponds to the crack spacing as illustrated in Figure 3.11. This method leads to a characteristic crack spacing  $s_{ch}$  for each parameter set investigated on the basis of the elastic strain energy and the assumption of an initial crack length of  $100 \mu m$ . Since only the relative changes of  $s_{ch}$  are regarded for the life time analysis discussed in Section 4.7 these assumptions are appropriate to study the impact of the driving force on the initial spacing and service life qualitatively.



**Figure 3.11:** In order to find the characteristic minimum spacing of a parameter set the width of a model containing a  $100 \mu m$  long crack is increased until  $\Delta K_{eff}$  reaches the threshold value  $\Delta K_{th}$ .

### 3.3.6 Data Processing II

As mentioned above, data processing is already implemented in the script. The output consists of two text files which can simply be copied to Origin or Excel. One file records the K factors of all cracks occurring in the modeling chain and the other one contains all data on the propagating crack as shown in Table 3.4. The column "Factor" refers to

**Table 3.4:** Format of the main output file.

$a$	$\sigma_{min}$	$\sigma_{max}$	R	Factor	$K_{max}$	$\Delta K_{eff}$	Instability	$N_i$	$\Sigma N_i$
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.

the internal transformation of  $K_{max}$  into  $\Delta K_{eff}$  by using a function stored in the Data Module. This function receives R as input and returns a transformation factor. The elements of column "Instability" are either 0 or 1, where a 1 means that the array has

gone through an instability, viz. cracks stopped at the corresponding crack lengths. As previously mentioned, the term instability refers to the configuration of the array and not to the fracture mode.

# Chapter 4

## Results

### 4.1 Temperature Distribution

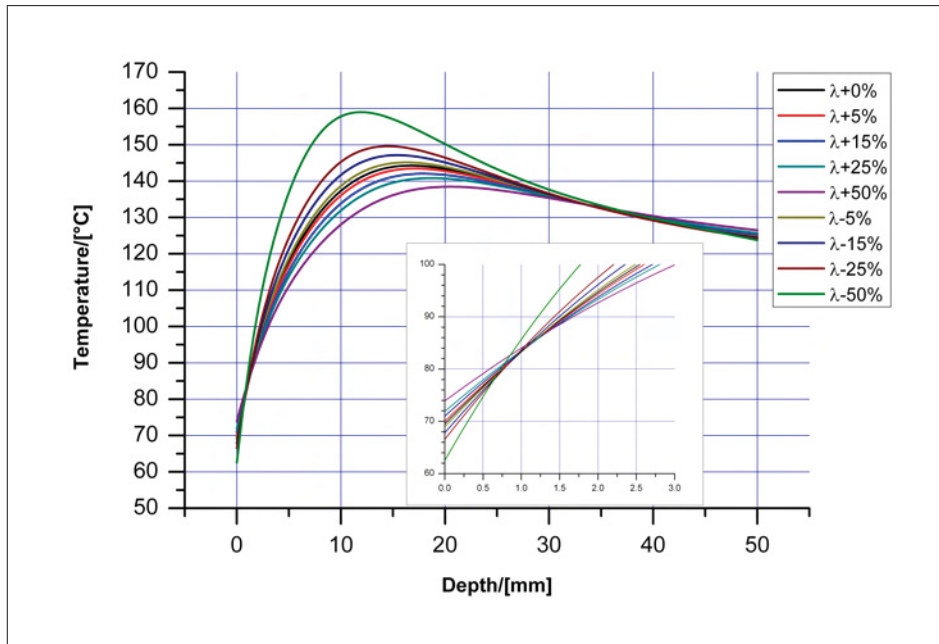
Figure 4.1 shows temperature distributions after spraying for different values of  $\lambda$ . The influence of  $\lambda$  on crack growth is assessed for two points, i.e. the surface region and the point where tension changes to pressure. The temperature gradient at the surface is the driving force for crack propagation in this region. A larger thermal conductivity leads to higher surface temperatures and hence to smaller gradients. This is favorable with regard to crack initiation and growth in the early stages of the formation of the crack array. Smaller gradients are a consequence of a larger heat flux. This comes along with decreased temperature maxima which are also shifted to larger depths. The position where the temperature exceeds the temperature after preheating, 120°C in this case, marks the point where the stresses change from tension to pressure. The cracks in a crack network usually do not reach that far into the die but the positions where stresses change to pressure are a measure of the maximum range of cracks which is especially important for single crack propagation. Because a larger value of  $\lambda$  increases the length a single crack can reach, smaller values of  $\lambda$  are favorable for later stages of crack growth. In order to prove these assumptions single crack propagation for three different values of  $\lambda$  is investigated, see Table 4.1. The simulations confirm the assumed relations of  $\lambda$  with propagation rates at the surface and maximum penetration.

Nevertheless, it can be said that the influence of  $\lambda$  in the earlier crack growth regime is the more significant one. In case of crack arrays the maximum crack length is not reached before service life has ended which, in turn, is primarily dependent on the speed of crack propagation at the surface. Thus, service life can be gained at the surface and nowhere else. The appearance of single cracks has to be avoided since they are likely to cause early

**Table 4.1:** Single crack propagation for three different values of  $\lambda$ . An increase in thermal conductivity causes lower propagation rates at the surface but larger maximum depths.

$\lambda$	-25%	$\lambda_{standard}$	+25%
$\frac{da}{dN}$ in a depth of 0.5 mm/[mm/Cycle]	0.001570	0.000799	0.000398
Depth after 100000 cycles/[mm]	7.25	8.25	9.25
Maximum depth/[mm]	8.25	9.75	10.75

failure. Other concepts are used to avoid single cracks and thus  $\lambda$  should be optimized with regard to crack growth in crack arrays. The formation of residual stresses which are significant for the initiation and growth of cracks (see Leindl [7]) further emphasizes the importance of the surface region.



**Figure 4.1:** Temperature distributions for various parameter variations. Larger values of  $\lambda$  cause smaller temperature gradients but also shift the point where stresses change from tension to pressure to larger depths, i.e. the intersection of temperature curves with the background temperature (120°C).

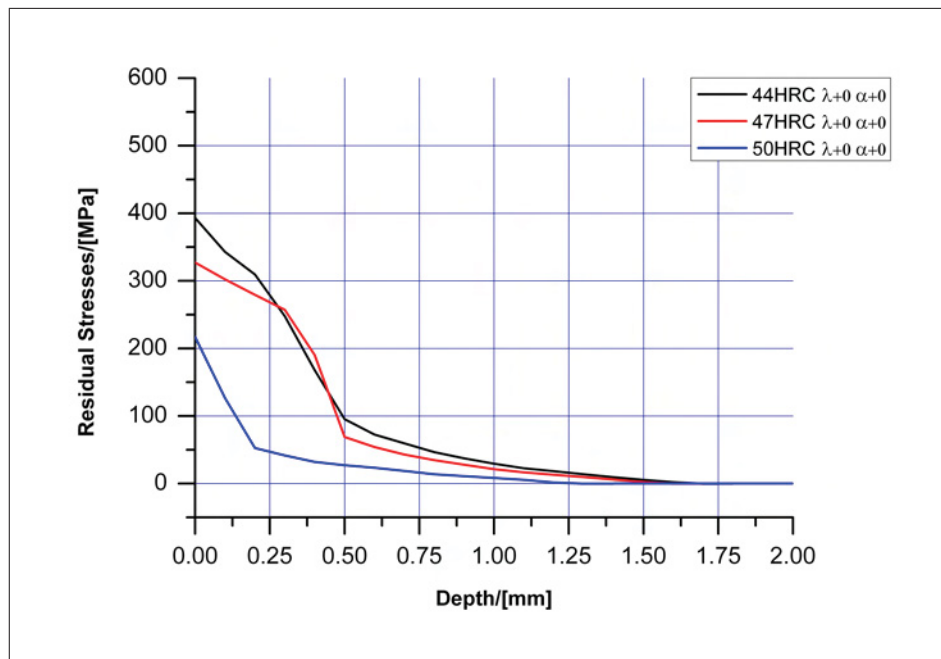
## 4.2 Formation of Residual Stresses

A wide range of experiments was carried out on X38CrMoV5-1 hot work tool steels subjected to various heat treatments in order to vary its hardness. This was done in the work of Ecker [4] whose material model provides the necessary tool to compute residual stresses in the present diploma thesis.

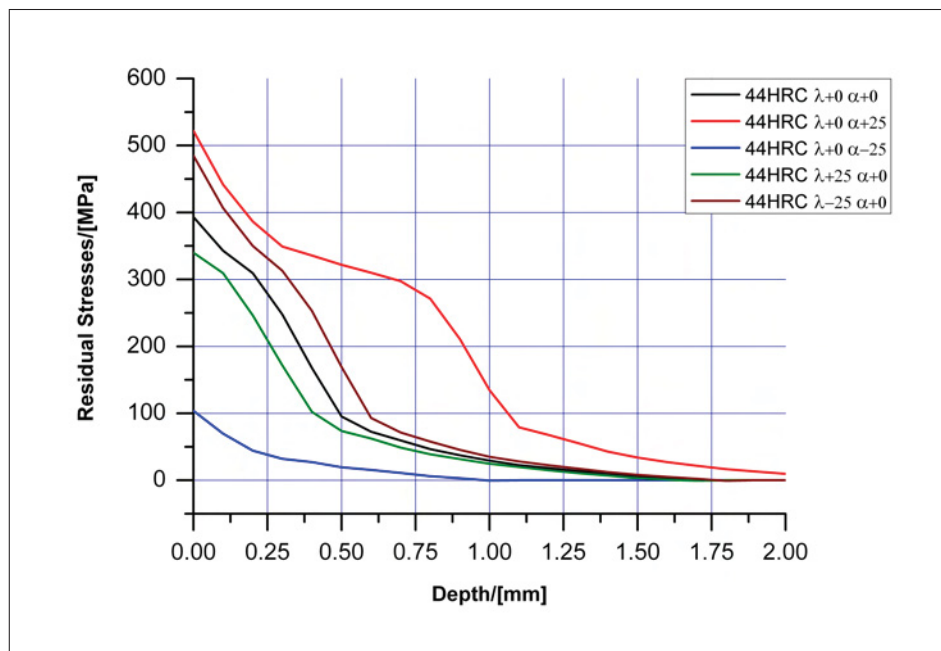
The shape of the  $\sigma_{res}$  distribution after 100 casting cycles is shown in Figure 4.2 for three different hardnesses, i.e. three different heat treatments. This material parameter is essential for the simulation of the material response to cyclic thermal loading because it affects all aspects of plasticity. The general shapes of the curves vary since the material responses vary due to the changes in the microstructure during heat treatment. Based on general observations, the residual stresses increase with the the plastic strain amplitude of the material. This statement is confirmed by Figures 4.2 and 4.3. Figure 4.3 shows the distributions for a range of parameter variations ( $\alpha$  and  $\lambda$ ) of a X38CrMoV5-1 with 44 HRC. Thermal expansion  $\alpha$  and thermal conductivity coefficients  $\lambda$  do not change the flow curves of the material.  $\lambda$  changes the temperature distribution which determines the expansion of the material given by  $\alpha$ . As was already discussed in [7] and [4]  $\alpha$  has much more impact than  $\lambda$ . However,  $\lambda$  can better be influenced by alloying and thus offers a wider margin.

This part of the present work involves a parametric study with the aim to get an appropriate amount of data to find a fit function which is supposed to allow engineers to quickly estimate  $\sigma_{res}$ . The stress values at the surface for a range of parameters is depicted in Figure 4.4. The bottom plane represents the parameter space and the other two planes show 2D diagrams  $\sigma_{res}$  vs  $\lambda$  and  $\sigma_{res}$  vs  $\alpha$ . As expected, decreasing values of  $\lambda$  and rising values of  $\alpha$  cause larger residual stresses. Below an  $\alpha$  of 75% of the standard value, the formation of residual stresses becomes unlikely. Corresponding data for hardness levels of 47 HRC and 50 HRC can be found in the appendix.

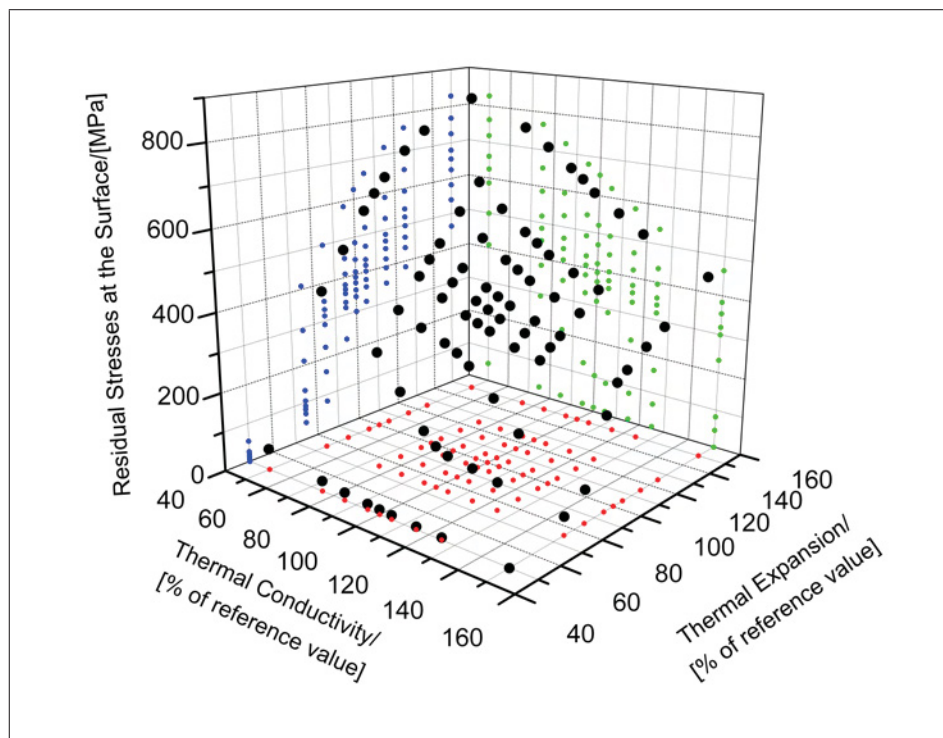
The data is fitted with a cubic polynomial given by Equation 4.1. Its precision is good enough for estimates, especially in the important region not too far away from the reference material. Further away deviations are larger and at 50% of the standard value for  $\alpha$  the polynomial gives negative stresses. In this region the residual stresses are 0 except at small values of  $\lambda$ . Figure 4.5 shows the polynomial and the corresponding coefficients. The values for  $\lambda$  and  $\alpha$  have to be inserted in percent.



**Figure 4.2:** Distribution of residual stresses after 100 casting cycles in X38CrMoV5-1 hot work tool steels for three different heat treatments as initial states before cyclic loading.

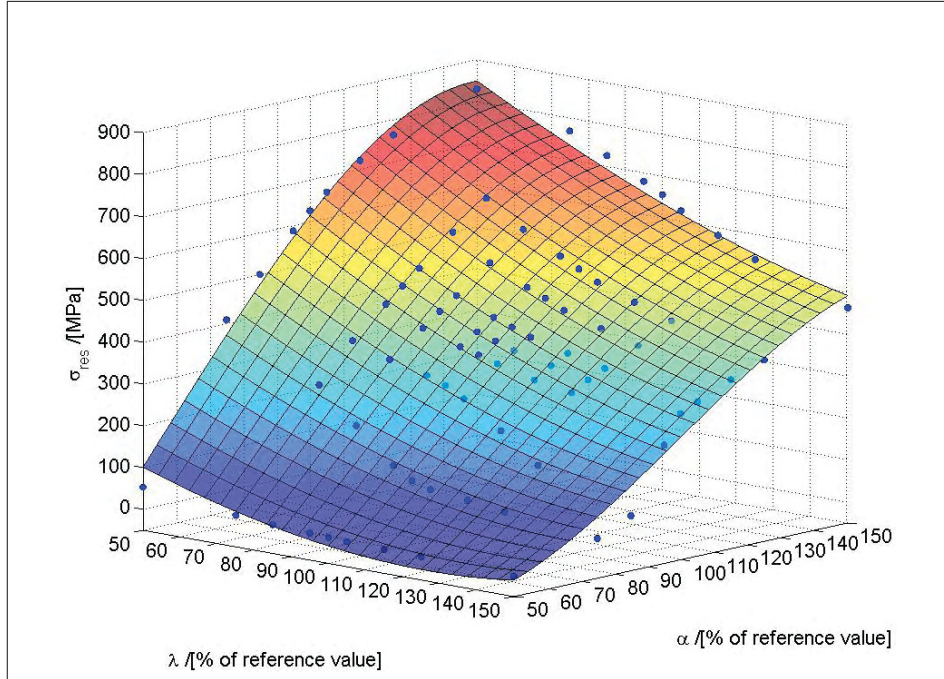


**Figure 4.3:** Distribution of residual stresses after 100 casting cycles in a X38CrMoV5-1 hot work tool steel with a hardness of 44HRC for different material parameters.



**Figure 4.4:** Overview of residual stresses at the surface of a X38CrMoV5-1 hot work tool steel with a hardness of 44HRC after 100 casting cycles for different material parameters.





**Figure 4.5:** Polynomial fit of data shown in Figure 4.4.

$$\sigma_{res} = a_1\lambda^3 + a_2\alpha^3 + a_3\lambda^2\alpha + a_4\lambda\alpha^2 + a_5\lambda^2 + a_6\alpha^2 + a_7\lambda\alpha + a_8\lambda + a_9\alpha + a_{10}$$

$$\vec{a} = \begin{pmatrix} 0.0000417780370985064 \\ -0.000428286703569801 \\ -0.00006789748905627 \\ 0.00055257973049667 \\ 0.0121419541794962 \\ 0.0448385372432628 \\ -0.121237574248307 \\ 0.459024312533427 \\ 13.1270980388396 \\ -430.296239561614 \end{pmatrix} \quad (4.1)$$

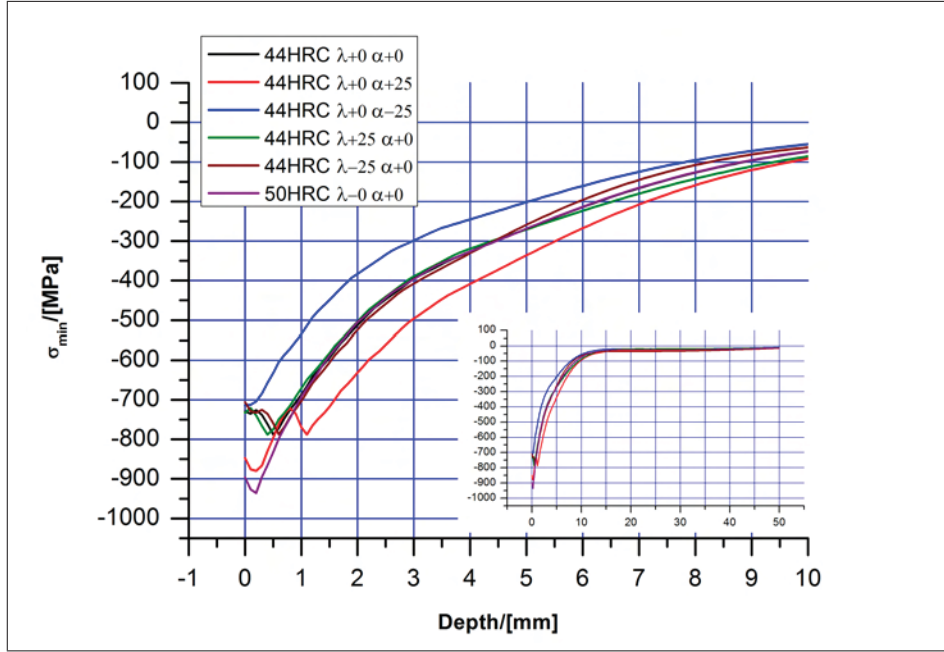
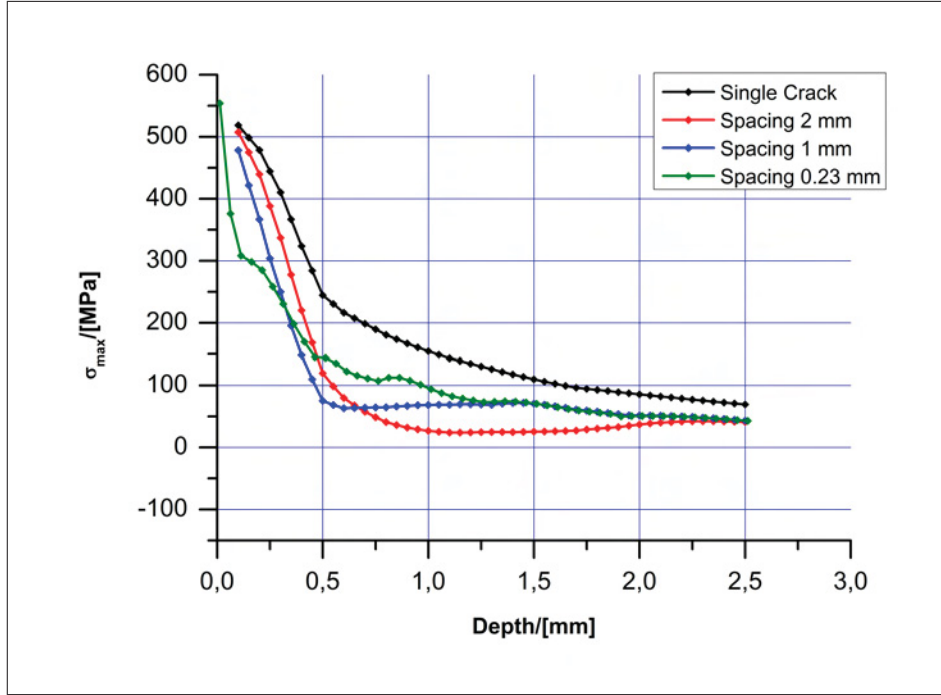


Figure 4.6: Distributions of  $\sigma_{min}$  for different material parameters.

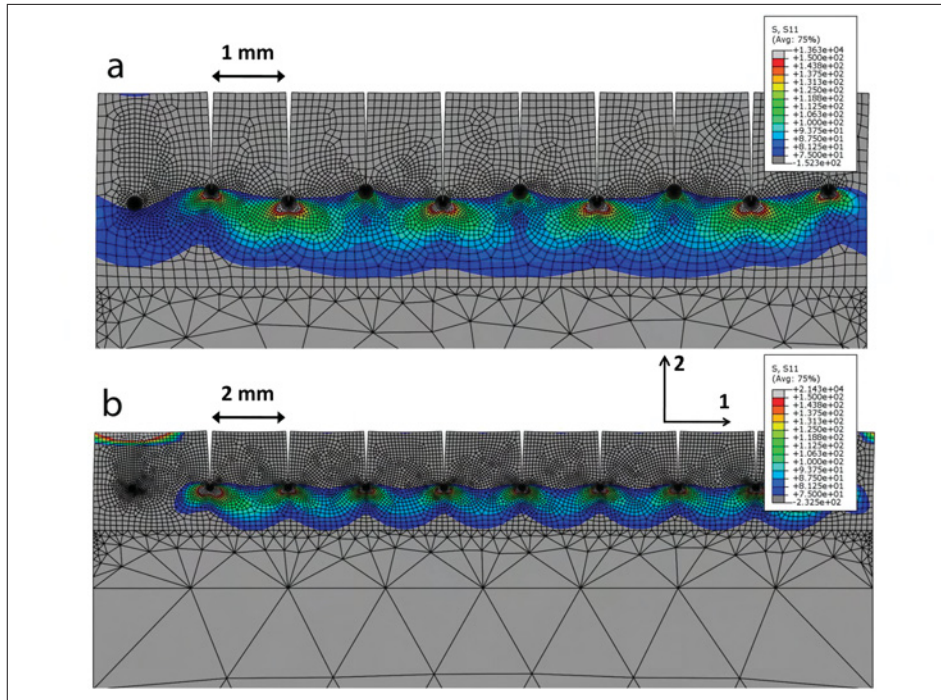
### 4.3 Stress Ratio

The distributions of  $\sigma_{min}$  for different material parameters are depicted in Figure 4.6. It is evident that the distribution is influenced mainly by residual stresses and  $\alpha$ . The residual stresses are dominated by hardness and  $\alpha$ .  $\lambda$  does not have much impact. The distributions of  $\sigma_{max}$  are depicted in Figure 4.7. As long as no instabilities in the configuration occur the curve progressions are similar for different initial configurations. With decreasing crack spacing  $\sigma_{max}$  reaches lower values which is not surprising after the discussions in the last section. In case of instabilities  $\sigma_{max}$  jumps to larger values as expected and it stays at an elevated level with respect to a corresponding curve with no instabilities. This can be explained by the stress concentration areas of the neighboring cracks arrested shortly before. Since the propagating cracks need to pass through the stress fields of the neighboring cracks their crack tips are subjected to higher loads. Curve number 4 represented by the green line in Figure 4.7 was taken from the simulation of a crack array starting from the lowest possible crack length and crack spacing. This curve shows instabilities at larger levels than curve number 2 (red line) which follows stable behavior. This can be explained by the way  $\sigma_{max}$  is determined. Two characteristic  $C - Models_{\sigma_{max}}$  are shown in Figure 4.8. The model in Figure 4.8a just went through an instability. Crack number 1 is kept shut in each model as discussed in Section 3.3.1 and the same range of



**Figure 4.7:** Distributions of  $\sigma_{max}$  for various initial configurations.

$\sigma_{11}$  is displayed. It can be seen that the area of the crack tip 1 is influenced by its neighbors which have already stopped whereas the corresponding area in the other model (see Figure 4.8b) with a larger initial spacing and no instabilities does not show this behavior. Thus, larger stresses in combination with smaller spacings are possible. Note that there is a crossover of the blue and the red line before an instability occurs (Figure 4.7) due to the fact that the stress fields around the tips of the cracks on the left and on the right of the closed crack have already become large enough to influence the crack tip region of crack 1 in the configuration corresponding to a spacing of 1 mm. It might seem confusing that some cracks turn out to stop at comparatively large stress levels (as represented by the fluctuations of the green curve indicating instabilities) as opposed to the continuously and uniformly growing configurations represented by the red or blue curve where the stress level is considerably lower. This can be explained simply by the different calculations of  $\sigma_{max}$  and  $K_{max}$ . The surface areas of the closed cracks in Figure 4.8 are subject to larger stresses since they are less shielded because the crack is missing. In Figure 4.8b these stresses are far larger than in the other model which shows larger  $\sigma_{11}$  in the tip area. In the *C – Models* the surface stresses pull the crack flanks apart and influence the stress intensity largely. This is why  $\sigma_{max}$  in *C – Models<sub>max</sub>* is not significant for the stability criterion.



**Figure 4.8:** Two characteristic models. a:  $C - Model_{\sigma_{max}}$  for initial spacing 1 mm; b:  $C - Model_{\sigma_{max}}$  for initial spacing 2 mm

Figure 4.9 shows R-Distributions for various material parameters which give rise to the following conclusions:  $\alpha$  and hardness have great influence on the residual stresses especially in the surface area. In greater depths the R-distributions coincide with the curve for standard parameters. This might seem suspicious because in case of different values for  $\alpha$  the stresses due to the temperature distribution are also different even at greater depths. Since stresses in the tensile and pressure regime are affected exactly the same way this effect cancels out when the stress ratio is calculated. Because X38CrMoV5-1 steels with 44 HRC or 50 HRC only differ in their plastic response to external loads their R-distributions only differ in the area where residual stresses are present. The impact of  $\lambda$  is significant in areas deeper than the range of residual stresses. This is due to the fact that the slope, the maximum value and the position of the maximum value of the temperature distribution determine all stresses not related to residual stresses.

## 4.4 Crack Shielding

The purpose of this study is to illustrate the impact of crack shielding on the stress intensity factors in a crack array. The script starts a chain of models each containing

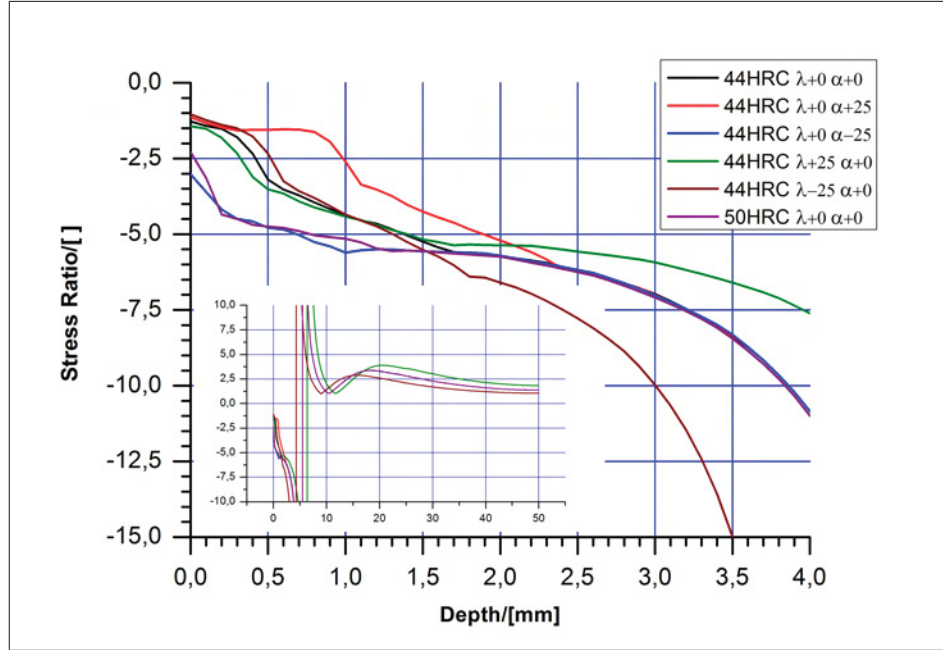
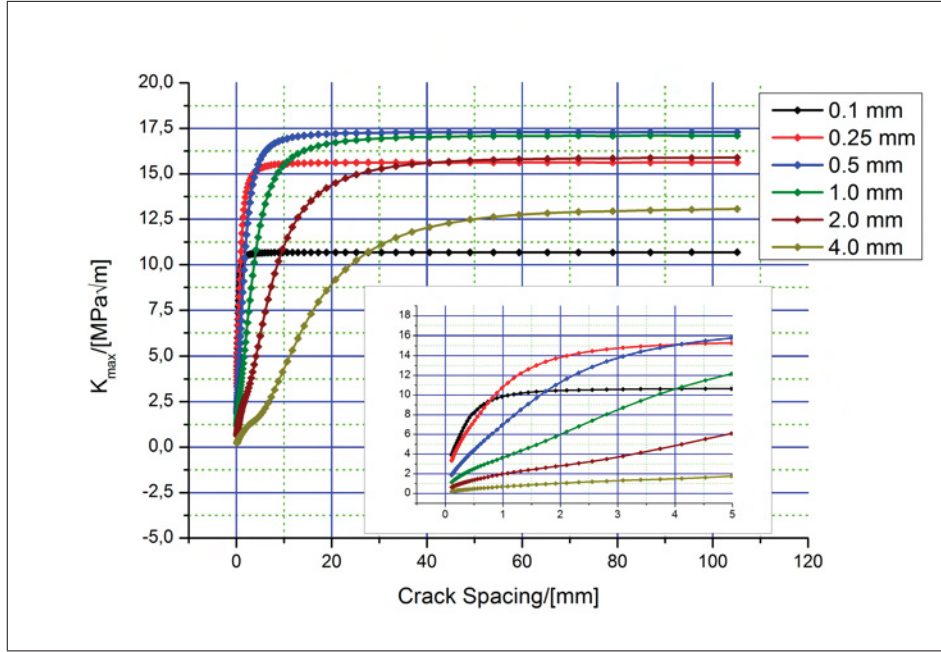


Figure 4.9: Stress ratios for different datasets.

a single crack. The width of the models is increased steadily up to 100 mm. Due to the periodic boundary conditions the width of the model corresponds to the distance between neighboring cracks. After reaching this value, a new chain starts with a longer crack and this sequence is repeated for a set of crack lengths. The stability criterion is switched off in the settings of the Control Script, since it is not required for this simple purpose. It would be possible to do the same task analytically based on information from the Stress Intensity Factors Handbook [37] which gives analytical solutions for a range of crack configurations, however, the manual calculations would be tedious because of the loading in this case leading to complex stress distributions.

The gathered information is supposed to give an impression, how far the shielding effect reaches and what settings are needed for a study on single crack propagation. The curves obtained are similar to Figure 2.21 but they consider the specific loading conditions in aluminum pressure casting dies. The shielding takes effect within a length scale at least twenty times the crack length. Since the curve approaches a constant value at far distances a factor of 15 is enough for studies on single crack propagation. A factor of 10 can still be regarded sufficient if computation time strongly limits the model size. It should be considered, however, that between a factor 10 and 20 a difference of  $1 \text{ MPa}\sqrt{\text{m}}$  in stress intensity can occur which considerably affects the computed life time due to the exponential relation between  $\Delta K_{eff}$  and the corresponding crack propagation rate.



**Figure 4.10:** The influence of crack shielding covers a distance of about 20 times the crack length.

## 4.5 Evolution of Crack Arrays for Standard Parameters

The evolution of crack arrays in dependence on material parameters and initial configurations is the core study of this thesis. This section covers in detail the evolution of crack arrays depending on the initial crack spacing for the case of a X38CrMoV5-1 hot work tool steel with a hardness of 44 HRC. The next section deals with parametric studies linking service life with the material parameters  $\lambda$  and  $\alpha$ . In both cases the whole functionality of the Control Script is made use of where data of all preceding studies is required. Based on an initial crack configurations the script starts a series of dependent C-Models. All cracks are assumed to initiate at the same time and therefore the initial unit cell consists of one crack only. In each C-Model the crack propagates by an increment of  $50 \mu m$  where the functionality summarized in Table 3.3.4 is implemented. After the analysis of each C-Model the Control Script decides the design of the following model based on the stability criterion. In case  $\Delta K_{eff}$  reaches or falls below  $\Delta K_{th}$  the unit cell is doubled in width and number of cracks. From all the cracks in the unit cell always the first one is viable and therefore propagates, whereas the others stay at the same length. In the course of the simulation the width of the C-Models and the number of cracks increases until an abortion criterion is satisfied.

## Preliminary Studies

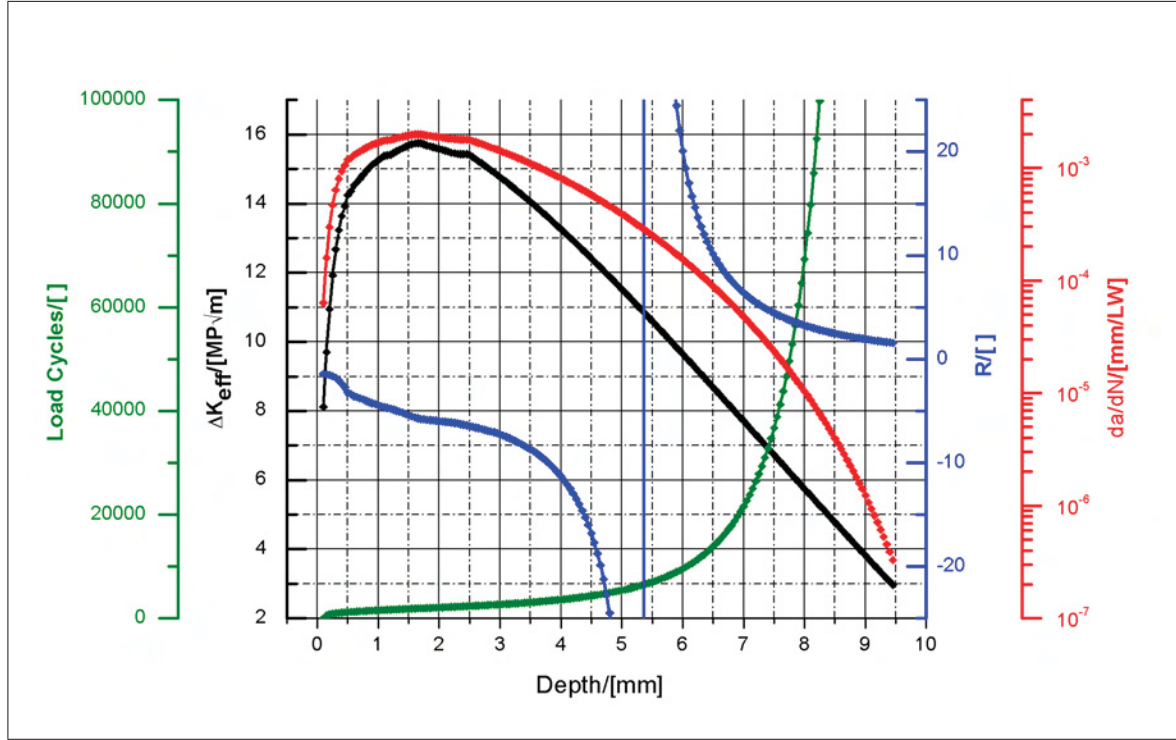
In a first step the functionality of the script and the limits of the C-Model are tested. For this purpose two Control Scripts with different settings are started in order to study two extreme conditions, viz. very large and very small crack spacing. Since real configurations are situated between these extremes it is essential to know what happens at each boundary. The first script conducts a study on a crack not subjected to the influence of crack shielding (see Figure 4.11). Based on the knowledge of the last section the model width for the study on single crack propagation is set to 100 mm. A variable model width would be desirably but this study is confined to a fixed width to avoid influences of the mesh. The initial crack length  $a_0$  is set to 0.1 mm. The essential output is depicted in Figure 4.11 which clearly shows that a configuration like that would be fatal for a die. The stress intensity factor does not show any influence by other cracks and results entirely from the combination of loading and crack length. It would reach a depth of 2.5 mm in less than 2400 casting cycles and a final crack length of 9.5 mm. Since the model width is set to 100 mm which is already close to 10 times the final crack length some shielding influence may be involved at larger crack lengths. The propagation rate  $\frac{da}{dN}$  in the final increment gets close to  $3 * 10^{-7} \frac{mm}{Cycle}$ . This is the order of magnitude of a single Burgers vector which marks the minimum crack extension and the limit for crack growth [38]. At a depth of 5.4 mm R changes sign because  $\sigma_{max}$  drops below zero. This does not have any effect on crack closure<sup>1</sup>. The maximum effect of crack closure still prevails ( $\Delta K_{eff} = K_{max}$ ).

In reality configurations similar to the one in this study emerge and cause failure because single cracks hit the cooling channels. These configurations do not appear on even parts of the dies but in areas with intricate geometries. There are other cracks as well but just one of them prevails and behaves like a single crack without shielding. The next study is supposed to give an impression of the slowdown effect of shielding.

The second script is set to find the smallest viable crack with the corresponding minimum spacing for standard parameters. The result is an array consisting of cracks with a length of 13  $\mu m$  and a spacing of 230  $\mu m$ . The spacing is computed by the method described in Section 3.3.5. Crack growth of smaller cracks cannot be described by the C-Models since small crack behavior significantly differs from the regime approximated by Paris equation. Small cracks grow even below  $\Delta K_{eff}$  and at faster rates than expected [20].

Based on this initial configuration the script starts to simulate the evolution of the array as illustrated in Figure 4.13. This study shows the influence of crack shielding which causes several instabilities. In the beginning  $\Delta K_{eff}$  rises due to the combination of load-

<sup>1</sup>A positive R would have effect if  $\sigma_{min}$  exceeded 0.

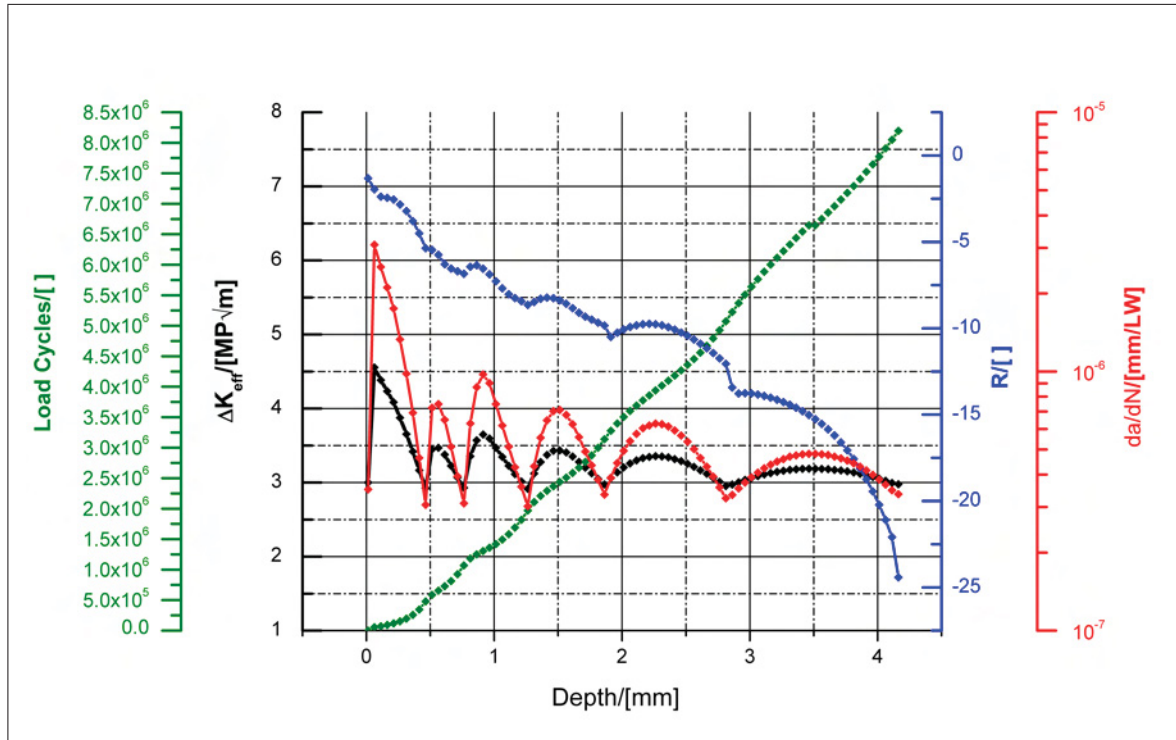


**Figure 4.11:** Data on single crack propagation.

ing and crack length similar to the previous configuration. After the next propagation increment shielding is dominant and the curve progression is headed towards the first instability (see Figure 4.12). It seems like a more narrow optimum spacing might evolve between the first two increments as described by Jenkins (Section 2.8). However, this will not happen under the present loading conditions since cracks shorter than the main cracks are exposed to extreme shielding effects (see Figure 3.5). The stress intensity of cracks arrested due to an instability drops immediately by a significant factor. Crack propagation continues until the crack is no longer viable. Again the propagation rate  $\frac{da}{dN}$  in the final increment approaches  $3 * 10^{-7} \frac{mm}{Cycle}$ .

Since crack growth takes place near the fatigue threshold for the most part, an extraordinary fatigue life may be expected. The dies are replaced every 100000-150000 cycles and according to the green line in Figure 4.12 the cracks in the array would never pass a length of 0.25 mm. A crack configuration with similar spacing, however, cannot be observed in aluminum pressure die casting dies even though the driving force concept predicts a crack distance of 230  $\mu m$ . This issue will further be discussed in Section 4.7.1. Crack growth in regime I as discussed in Section 2.4 is an issue in this case. The range of  $\Delta K_{eff}$  is almost completely in regime I and thus the estimations of load cycles is conservative because the



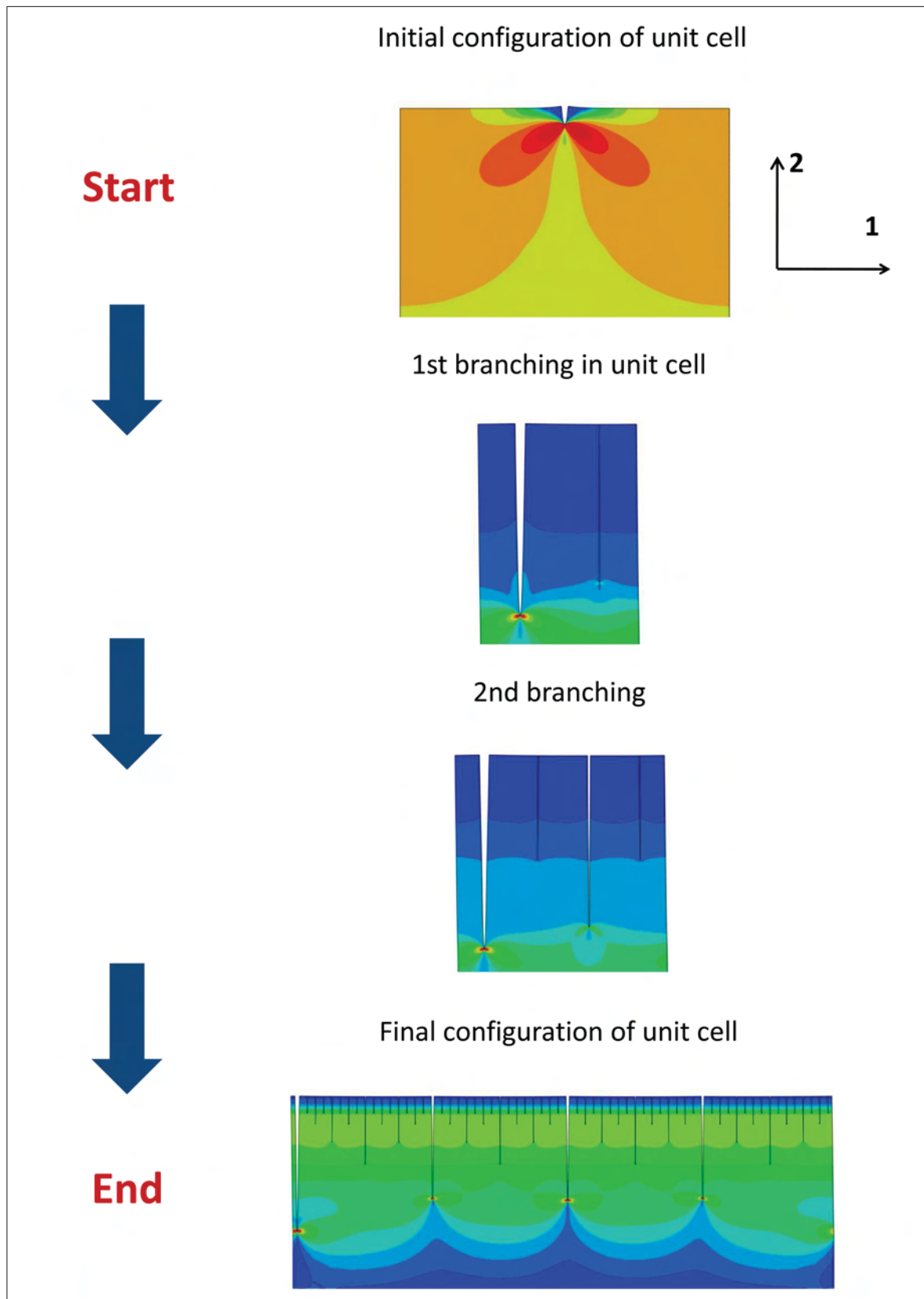


**Figure 4.12:** Data on a crack array with with the closest possible initial spacing.

uncorrected Paris equation overestimates crack propagation rates in regime I. It does not make sense to spend too much effort on applying the corrected Paris equation for a configuration that will not occur in reality. the correction only plays a role in the studies on real configurations with realistic spacings. It can be anticipated that this is not necessary, since one or two instabilities of the crack configuration do not cause much deviation in calculated life time based on the unchanged Paris equation.

Near threshold fatigue is not the only issue, as mentioned above, since it has to be checked whether the cracks are still behaving like small cracks. According to Equation 2.35 the initial configuration starts in the small crack regime but changes to the long crack regime within the first propagation increments. Thus corrections are assumed needless. Regarding single crack propagation, however, leads to a different result. The small crack regime would reach several millimeters into the die which is not realistic. The corrections discussed in Section 2.6 are derived on the basis of external loading conditions and where Equation 2.27 is applicable. However, in the present case of interacting cracks 2.27 is no longer valid. Apart from the estimations based on continuum mechanics it is possible to discuss the individual types of small cracks. According to the definitions in Section 2.6, the crack in the second study is microstructurally small since the grain size is about 25

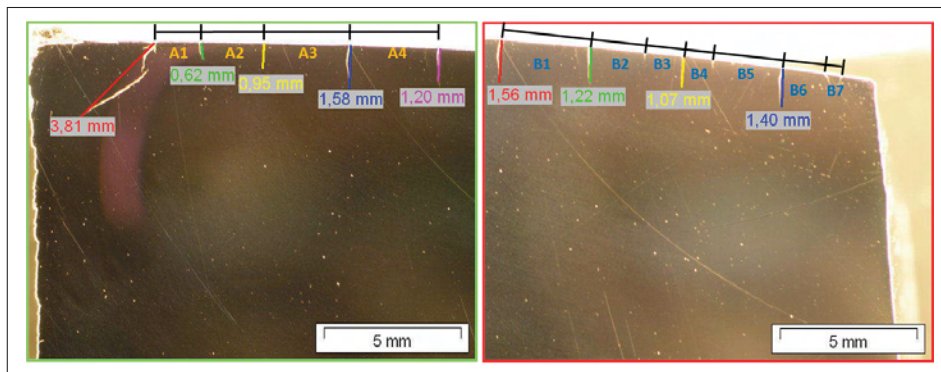
$\mu\text{m}$ . In the subsequent studies an initial crack length of 0.1 mm and long crack behavior right from the beginning will be assumed even though this might cause some error because 0.1 mm are less than 5-10 times the grain size and cracks of this length can still be microstructurally small. Plastic shakedown takes place up to a depth of 0.5 mm, between 0.5 and 1.7 mm elastic shakedown occurs. Due to this fact all cracks are mechanically small within the top 0.5 mm and from this depth on it would have to be checked, whether the cracks are mechanically small or long. The definition of mechanically small cracks comes along with the validity of LEFM. This will further be discussed in Section 4.8. The exact investigation of small cracks goes in line with studies on crack initiation. These topics are left to future work.



**Figure 4.13:** Scheme of modeling the development of a crack array. The contours show the normal stress distribution in 1-direction.

### Approximation of Service Life Time

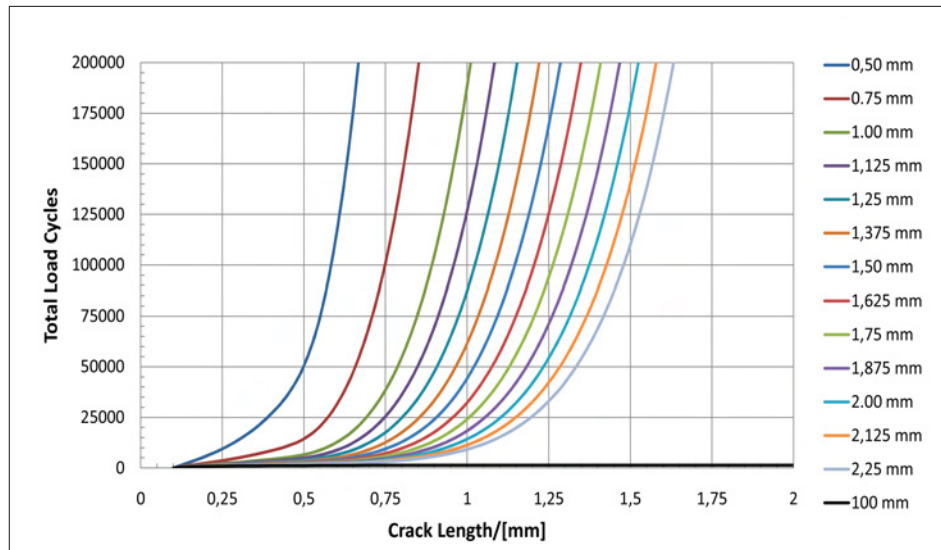
Conclusions from the last section suggest real configurations to be in between the two extreme conditions. In the course of the whole project the samples were taken from a die provided by industry after 130000 loading cycles. In nearly even portions of this die typical networks, as shown in Figure 4.14, were found and analyzed. The orientation of the cracks clearly follows the temperature gradient during casting and thus the arrays consist of parallel cracks except for positions at the outer edges. The average spacing based on these two arrays is determined to be 2.12 mm. In order to test the precision



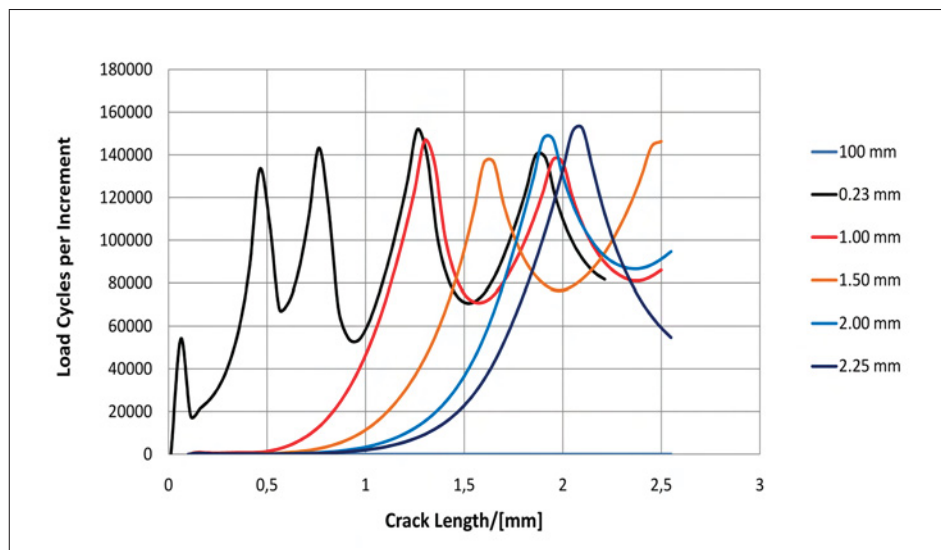
**Figure 4.14:** Crack arrays at two different positions of a die received from industry after service life was reached.

of the Control Script and the C-Model, a whole set of arrays is simulated with various initial spacings, starting from 0.5 mm up to 2.25 mm. The results are depicted in Figure 4.15. In this study a very good accordance between simulation and real observations is accomplished. The script based model predicts a maximum crack length of about 1.5 mm for an initial spacing of 2.125 mm. For subsequent studies on this configuration a value of 2.25 mm will be used since this leads to a value even closer to the observed maximum crack lengths of 1.56 mm and 1.58 mm, respectively.

Figure 4.16 gives an impression on how the number of load cycles is obtained and how the initial spacing and the occurrence of instabilities affect service life. The load cycles necessary for each crack propagation increment are obtained by integration of Paris equation in the current increment as discussed in Section 3.3.3. The total number of cycles required for a crack to reach a specific length is the sum of the cycles computed for the preceding increments. The closer the cracks are the sooner shielding takes effect which increases the number of load cycles. Instabilities on the other hand cause larger propagation rates and a decrease in cycles needed per increment. Still, the shielding effect prevails which keeps  $\Delta K_{eff}$  low and leads to the next instability soon after. The



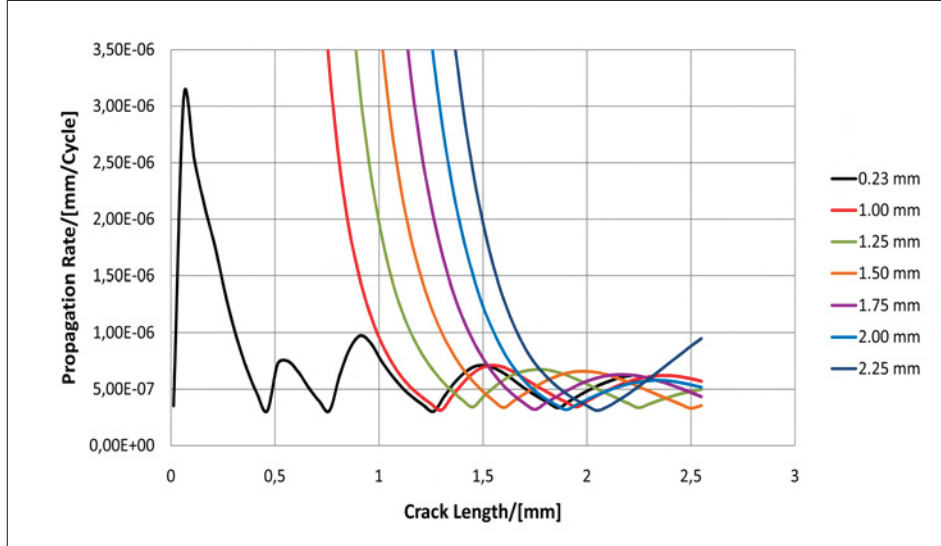
**Figure 4.15:** Progression of load cycles with increasing crack length for various initial spacings. This diagram shows that a small change of the initial spacing has a significant effect on service life.



**Figure 4.16:** Load cycles per increment for various initial spacings.

propagation rates are shown in Figure 4.17 which illustrates how fast surface cracks in real arrays grow in dependence on loading, crack length and shielding. After the first cracks have stopped  $\frac{da}{dN}$  stays at a level several orders of magnitude lower than in the single crack case (compare Figures 4.11 and 4.17). The black line was taken from the study on the closely packed array. The depth where cracks stop depends on the initial spacing and on

$\lambda$  and  $\alpha$  as will be discussed in the next section.



**Figure 4.17:** Propagation rates for various initial spacings.

## 4.6 Study on Service Life Time with Constant Initial Spacing

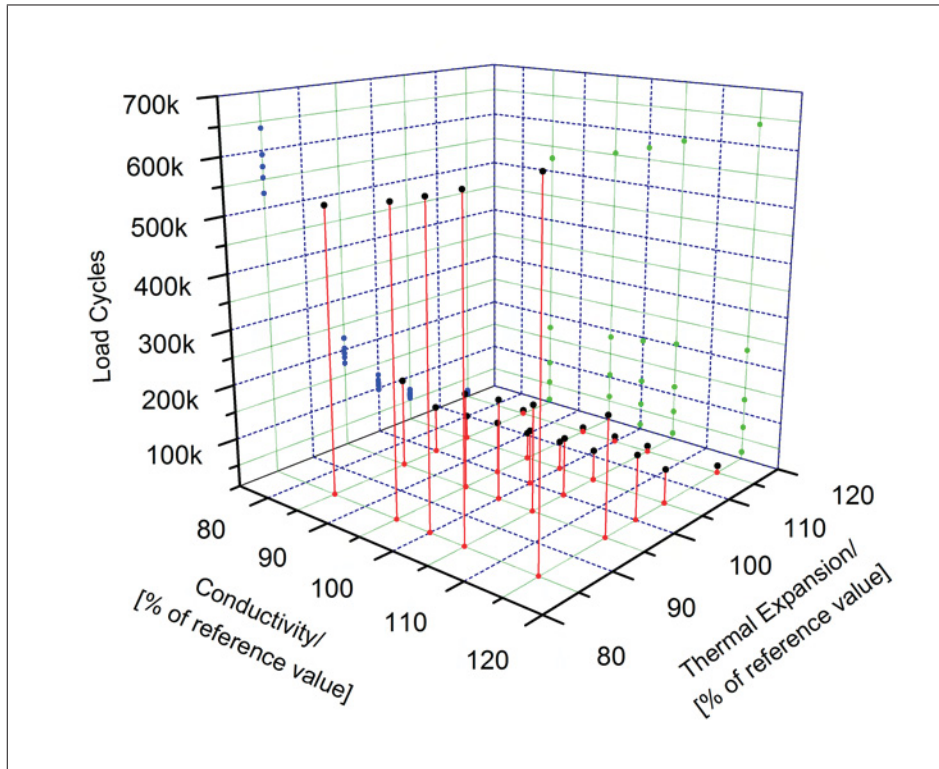
The basic idea behind this thesis is to implement the findings of Leindl [7] and Ecker [4] in an automated script together with further modifications and concepts. This script is supposed to allow studies on the specified problem to a greater extent as has been possible so far and in reasonable times. The study discussed in this section delivers the core result whereas the results in the next section are supposed to give some ideas on future work.

The ultimate goal of this thesis is to relate material parameters with the formation of crack arrays and thus service life. For this purpose, the study discussed in the preceding section was performed for 25 material parameter combinations for each of the three experimentally tested materials (Table 4.2). The other combinations are too far out of range for metallurgical treatments. The number of load cycles necessary for a crack to reach a specific length represents a measure for service life. Thus, the initial spacing of 2.25 mm obtained in the last section is used and the number of load cycles for a crack to reach a length of 1.5 mm is computed for all data sets. The script takes care of the creation of the C-Models, the computation of  $R$  and  $\Delta K_{eff}$ , the occurrence of instabilities and the integration of Paris equation. Since all the means necessary for this study have been

**Table 4.2:** Variations of  $\lambda$ ,  $\alpha$  and hardness used in the core study on a X38CrMoV5-1 hot work tool steel.

$\lambda$	$\alpha$	HARDNESS
-15%	-15%	44HRC
-5%	-5%	47HRC
+0%	+0%	50HRC
+5%	+5%	
+15%	+15%	

discussed and thoroughly checked in the previous sections, they will not be scrutinized again. Figure 4.18 shows the results for the case of a X38CrMoV5-1 with 44 HRC. As



**Figure 4.18:** Study on service life for a X38CrMoV5-1 hot work tool steel with a hardness of 44 HRC. The plot shows the number of load cycles until a crack length of 1.5 mm is reached. The initial crack spacing is 2.25 mm for all parameter combinations.

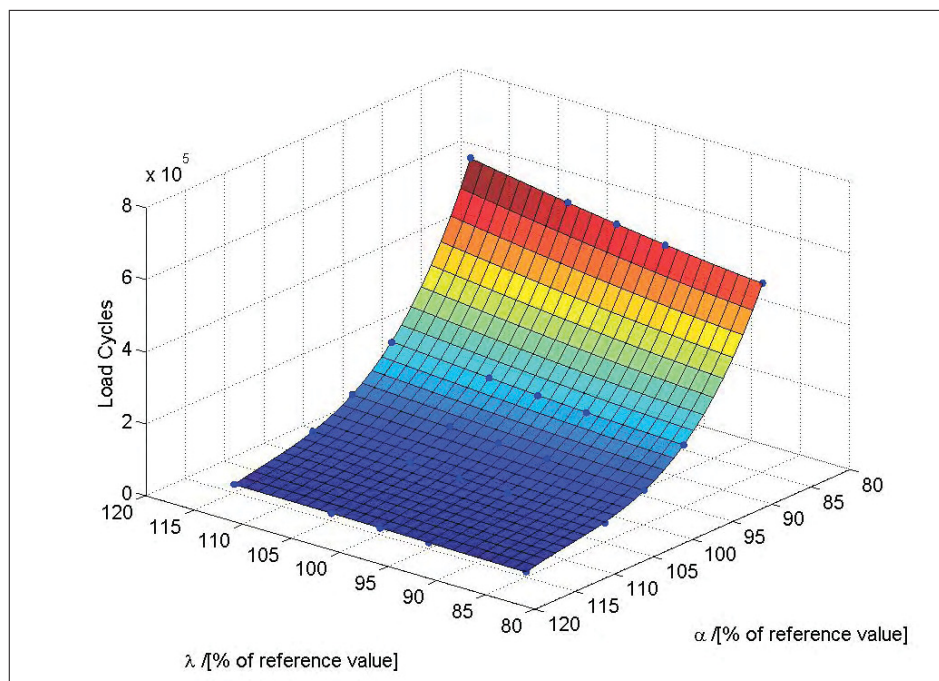
expected, an increase in thermal conductivity will increase service life whereas thermal expansion should be as small as possible for life time optimization. The impact of  $\alpha$  is by far greater but  $\lambda$  is easier to manipulate as mentioned above. The optimum combination is situated at the front corner corresponding to an increase of  $\lambda$  by 15% and a decrease of  $\alpha$  by 15% (for figures refer to Table 3.2). A closer view of the x-z and y-z plane in Figure

4.18 can be found in the appendix (Figures 6.5 and 6.6).

It was found that the cubic fit function used for in Section 4.2 suits the simple distribution in Figure 4.18 very well. The corresponding surface is depicted in Figure 4.19. Since service life depends inversely on the residual stresses, the axis are turned around to allow a view on the concave side of the surface. For the fit function  $\lambda$  and  $\alpha$  have to be inserted in percent.

The corresponding data for the steels with the hardnesses 47 HRC and 50 HRC were also fitted. The graphs can be found at the end of this section. Based on the results of the studies on service life an increase in hardness would be favorable. Higher hardness levels cause less residual stresses and thus a lower driving force for crack propagation. This conclusion is dangerous, however, since it does not consider all possible effects. Die materials with higher hardnesses are prone to single cracks which can cause early failure. For this reason those materials have already been ruled out as new die materials and a hardness level of 44 HRC was found to be a proper hardness. Furthermore, the effect of driving forces, e.g. the stresses within the die, on the initial spacing have not been taken into account so far. This issue, which is a possible way to explain the advantage of a material with a lower hardness, will be discussed in Section 4.7.2.

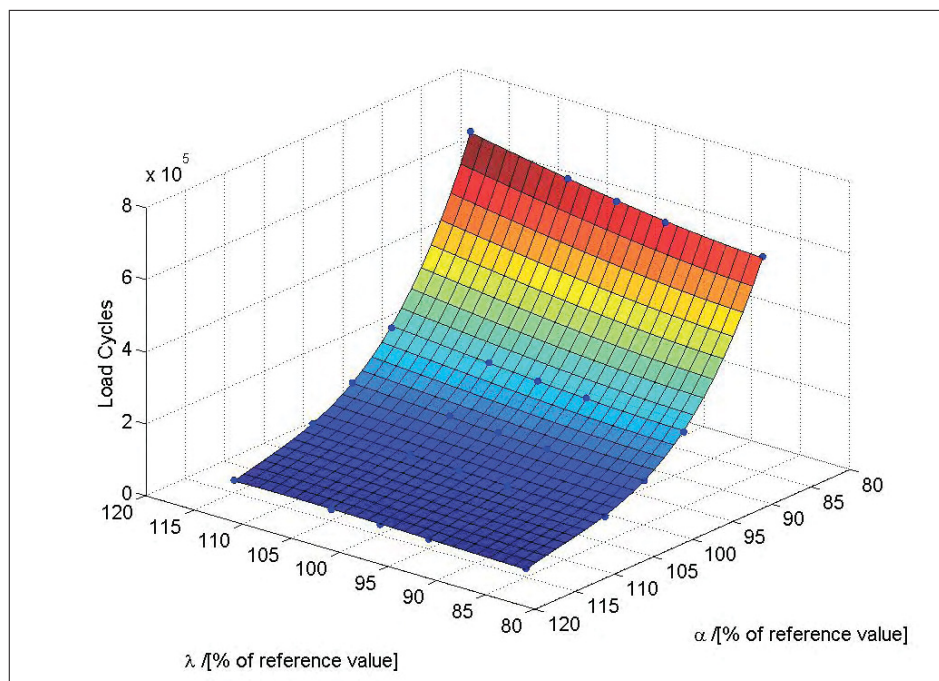




**Figure 4.19:** Polynomial fit of data shown in Figure 4.18 for 44 HRC. The axes were turned around for the sake of visibility.

$$N_{0.1-1.5mm} = a_1\lambda^3 + a_2\alpha^3 + a_3\lambda^2\alpha + a_4\lambda\alpha^2 + a_5\lambda^2 + a_6\alpha^2 + a_7\lambda\alpha + a_8\lambda + a_9\alpha + a_{10}$$

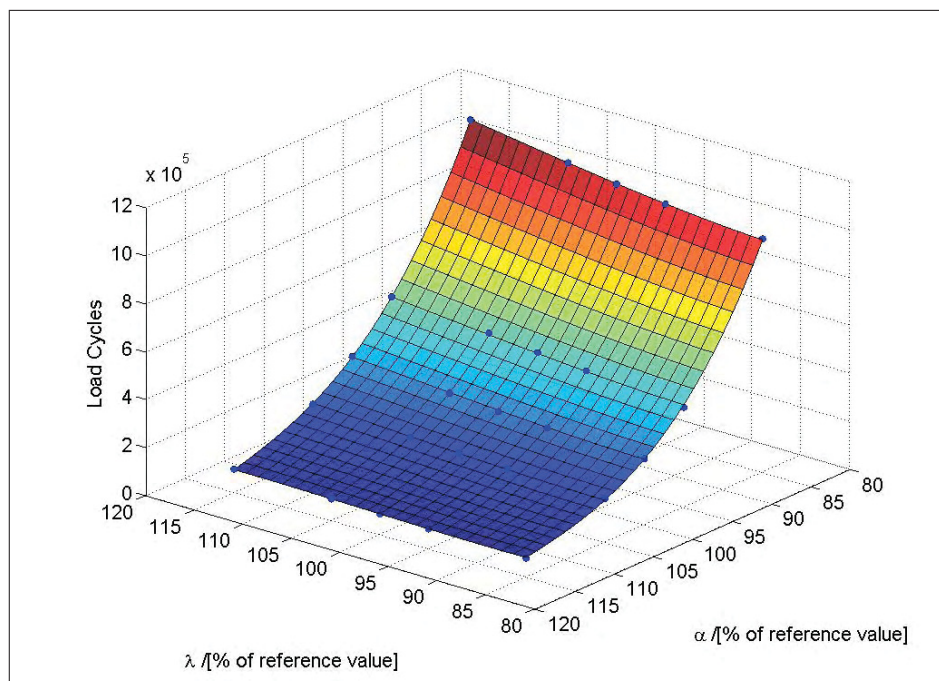
$$\vec{a} = \begin{pmatrix} -0.462447184209595 \\ -26.4792194875032 \\ -1.47435352672448 \\ 4.3281310842006 \\ 302.5322147049 \\ 8349.72443150749 \\ -687.193062516652 \\ 9404.60480419381 \\ -891131.377875977 \\ 32717386.4564178 \end{pmatrix} \quad (4.2)$$



**Figure 4.20:** Polynomial fit of data on service life for 47HRC. The initial crack spacing is 2.25 mm for all parameter combinations.

$$N_{0.1-1.5mm} = a_1\lambda^3 + a_2\alpha^3 + a_3\lambda^2\alpha + a_4\lambda\alpha^2 + a_5\lambda^2 + a_6\alpha^2 + a_7\lambda\alpha + a_8\lambda + a_9\alpha + a_{10}$$

$$\vec{a} = \begin{pmatrix} -1.05133164224793 \\ -26.062543903893 \\ -2.60589867458528 \\ 3.84401927699448 \\ 598.659263826688 \\ 8317.90514087566 \\ -361.565229199581 \\ -36962.9575094759 \\ -910741.681314945 \\ 35232112.4462281 \end{pmatrix} \quad (4.3)$$



**Figure 4.21:** Polynomial fit of data on service life for 50 HRC. The initial crack spacing is 2.25 mm for all parameter combinations.

$$N_{0.1-1.5mm} = a_1\lambda^3 + a_2\alpha^3 + a_3\lambda^2\alpha + a_4\lambda\alpha^2 + a_5\lambda^2 + a_6\alpha^2 + a_7\lambda\alpha + a_8\lambda + a_9\alpha + a_{10}$$

$$\vec{a} = \begin{pmatrix} -1.60460455042203 \\ -21.7377060170789 \\ -2.09151709927723 \\ 1.13716528022398 \\ 728.957657560104 \\ 7543.46667869434 \\ 40.4932568164583 \\ -68480.0783344633 \\ -888856.673040303 \\ 37203051.3966087 \end{pmatrix} \quad (4.4)$$

## 4.7 Influence of Material Parameters on Crack Growth with Variable Spacing

### 4.7.1 Possible Influences on the Initial Crack Spacing

In the previous study it was assumed that the initial crack spacing is constant for all data sets. Real surface shows preferred crack initiation sites. On the one hand this is caused by surface treatment techniques which lead to grinding grooves. These act as notches and thus cause stress concentrations. On the other hand it was shown that crack initiation depends on the microstructure of the material as schematically shown in Figure 4.22. Depending on plastic strain four different initiation regimes can be distinguished.

- Regime 1: Plastic strains are very low. Flaws beneath the surface act like notches and cause plastification. The areas around these flaws are initiation sites at a large number of load cycles. This behavior is typical for materials showing a two stage Wöhler diagram with two fatigue limits, one for the surface and one for the interior. In regime I the internal fatigue limit is predominant.
- Regime 2: Which one of the two limits prevails depends on the level of loading and the number of cycles the component is intended to withstand. At slightly larger plastic strains, crack initiation starts at the surface. Apart from surface flaws, grain boundaries of the former austenitic microstructure are preferred initiation sites.
- Regime 3: At intermediate levels of plastic strain, cracks also start to form at the borders between differently orientated martensitic regions within the previously existing austenite grains.
- Regime 4: At very high loads, cracks start to grow from the former austenite grain boundaries, from interfaces between martensitic regions, and from the borders of martensite laths or plates.

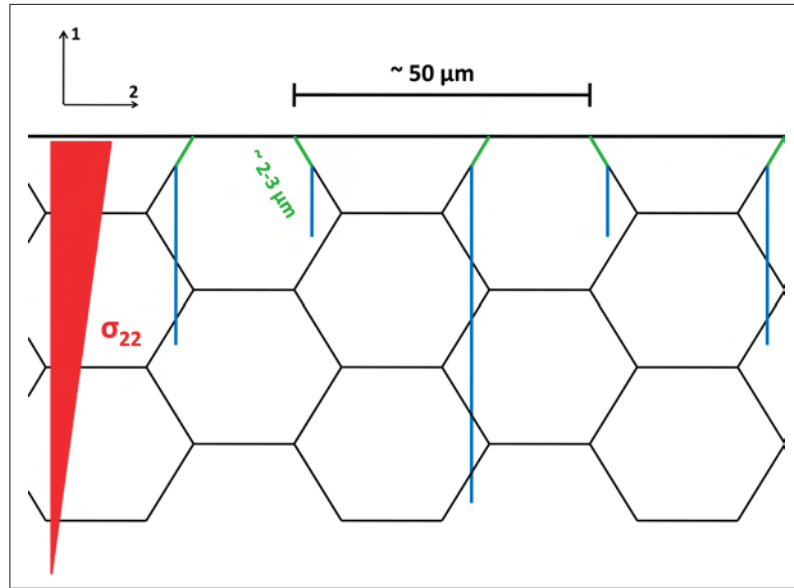
For the aluminum pressure casting dies regime 2 prevails. In theory, cracks initiate at former grain boundaries and grow to a few micrometers before they start to grow independently of microstructure. At first grain boundaries provide good conditions for initiation but further on the cracks will grow in the direction of the stress gradient. Crack spacing is therefore related to grain size which is about  $25 \mu m$  in the case of the investigated materials. Again, these assumptions are not fully correct. Not all these initiation sites are the same and hence the cracks will not initiate at the same time. Cracks appearing first

start to shield the initiation sites in the vicinity. Although it is observed that most cracks appear within a few thousand cycles, near surface crack growth is rapid and thus shielding quickly affects larger areas. Thus, potential initiation sites are already shielded when cracks could form at these positions. Even in case the influence is not strong enough to prevent crack initiation, the shielding crack will grow faster than the smaller ones and the shielding will increase. Hence, most of the surface cracks will already be shielded and stop before reaching a significant length. This is the reason why crack spacings significantly larger than the grain size are observed.

In the previous study it was assumed that the influence of the microstructure determines the initiation behavior at a certain loading condition. It was not considered that e.g. an increase in  $\alpha$  causes a significant increase in plastic strain which in turn could activate further initiation sites and influence the initial spacing. An experimental verification of these effects was nearly impossible at the time this thesis was written since no die materials existed showing such properties. Both the costs for the dies and for the manufacturing of trial alloys are by far too high to be able to use real dies for such a study. Furthermore, newly developed alloys would show a different stress-strain behavior. The development of the required material models would further increase the costs. This fact emphasizes the importance of the whole project because it is supposed to show the way to the next generation of die materials. The means developed and the results obtained so far are very instructive and promising for further research. If the approaches implemented in the investigations of the project are enhanced and new concepts are found it is possible to gain fundamental knowledge within weeks. The design of the modeling chain was designed for this purpose, i.e. to enable fast implementation of new concepts. A prospective concept will be discussed in Section 4.7.2.

#### 4.7.2 Study on Service Life Time with Variable Initial Spacing

In the study in Section 4.6 on service life it is assumed that the material parameters do not change the spacing. In order to get some insight into possible deviations, some estimates based on driving force are made. The initial crack length of a single crack is assumed to be  $100 \mu m$  and starting from a crack spacing of  $50 \mu m$  the distance is increased until  $\Delta K_{eff}$  reaches the threshold value  $\Delta K_{th}$ . In other words, one crack is assumed to be somewhere in an even section of the die and the script searches for the position of the next viable crack. This procedure is repeated for a set of material parameters and the results obtained are saved to a text file. By means of the Python script this file is designed to match a format that can be loaded as module back into the script. This enables the script in the



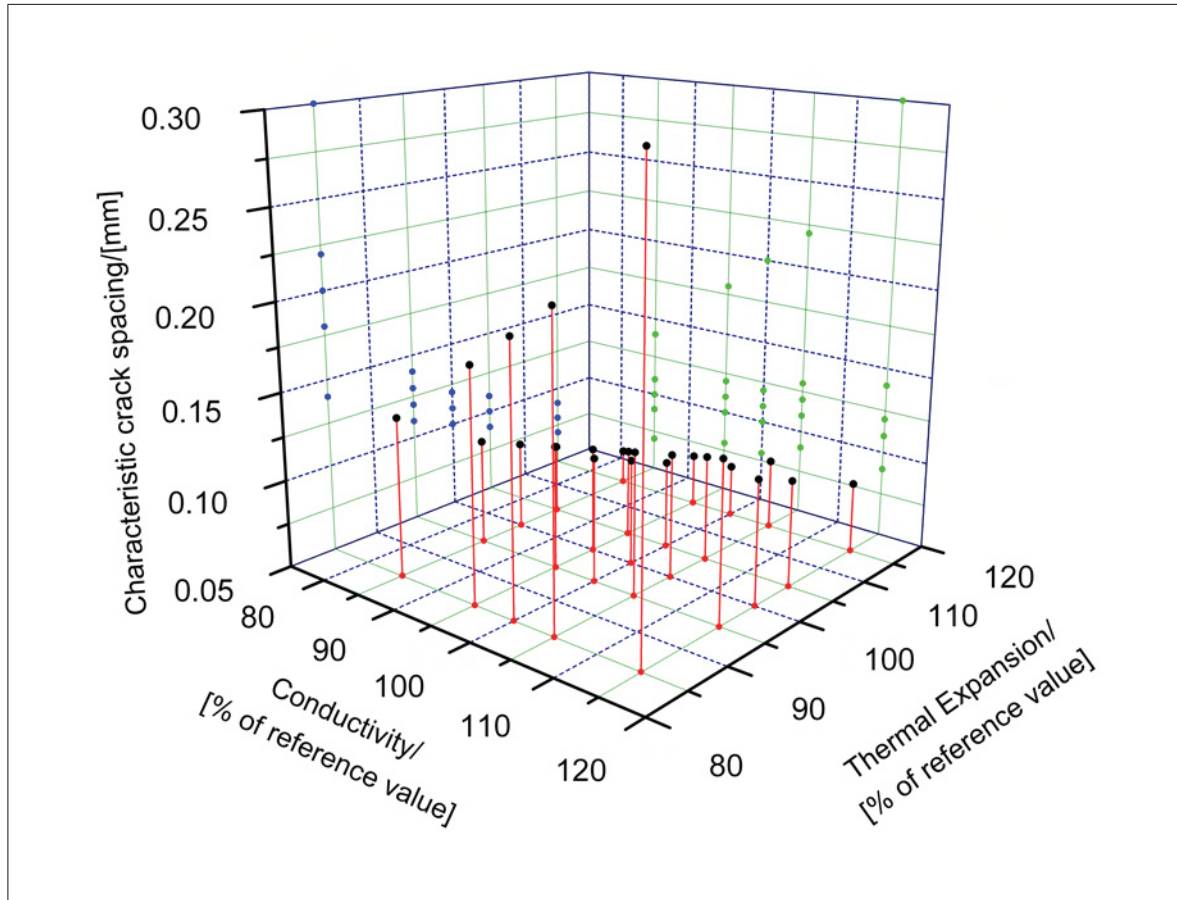
**Figure 4.22:** The influence of microstructure on initial crack spacing.

next study to access the initial crack configurations. The characteristic initial crack spacing  $s_{ch}$  is the smallest possible initial value for each material parameter set investigated. This approach is based on continuum mechanics, which assumes that every single spot on the surface is a possible site for crack initiation. Thus  $s_{ch}$  is related to the elastic energy stored in the die which in turn is related to the stresses. The interconnections between energy and strain have been discussed in Section 2.1. Simple relations are given by Equations 2.4 and 2.12. Despite the fact that direct relations between energy and stress based approaches have not been found for fatigue, they are still linked. Thus, this issue can be discussed in both terms, at least qualitatively. The larger  $\alpha$  the more energy will be stored in the material and the larger the stress level will be. Thus, cracks are viable in closer distances since they draw more energy from their surroundings. Every value of  $s_{ch}$  is characteristic for each combination of hardness,  $\alpha$  and  $\lambda$  because the parameters affect the energy available for crack initiation and crack growth. This information provided by continuum mechanics, however, is not the only factor determining crack spacing as discussed in Section 4.7.1. Other crucial factors are the microstructure and the surface of the material. Nevertheless, it is assumed that  $s_{ch}$  can be used as a measure for real crack spacing, based on the following thought pattern. When a die is exposed to thermal fatigue, the first crack will appear at an arbitrary position. Soon after more cracks will initiate somewhere at the surface of the die. All of these cracks behave like single cracks in that phase and do not feel each other. They will advance with fairly high propagation rates (see Figure 4.11) and each will soon shield an area several times its length. The

cracks initiating in the next few hundred cycles cannot form at arbitrary positions any more. They will fill the unshielded or not strongly shielded areas between the cracks that initiated before. This process itself is to be investigated by future work since so far all cracks have been assumed to form at the same time. This approximation is thought to be appropriate because the cracks are found to initiate within several hundred cycles. The results shown in Figure 4.15 suggest, however, that rapid crack growth at the surface already causes significant shielding. This and the influence of microstructure is supposed to be the reason why in case of standard material parameters the observed spacing is about 20 times larger than the calculated value of  $s_{ch}$ . Nevertheless, crack spacing can be related to the elastic strain energy stored in the system during spraying, the phase of a casting cycle, in which cracks propagate. It can be assumed that the initiation of the first cracks happens independently of each other, as has been described, for all parameter variations and stress levels. The situation changes, however, when later on cracks start to fill the areas between the first cracks. The number of cracks which can initiate and grow depends on the energy stored in the surrounding regions. This means that e.g. for low values of  $\alpha$  only 2 additional cracks fit in a gap whereas for large  $\alpha$ s maybe 10 viable cracks would fit in. This view might provide a basis for relating  $\alpha$  and  $\lambda$  to crack spacing.

Figure 4.23 shows  $s_{ch}$  for the parameter combinations used for the life time analysis in Section 4.6. For standard parameters for a X38CrMoV5-1 hot work tool steel with a hardness of 44 HRC this characteristic spacing was determined to be  $110\mu m$  and ranges from  $7\mu m$  up to  $300\mu m$  for the other data sets related to this hardness. The analysis in Section 4.6 was repeated two times. In both cases the value of 2.25 mm was decreased or increased by a factor determined from  $s_{ch,standard}$  and  $s_{ch}$  of the current parameter combination. In a first step the relative change  $\Delta s_{rel,i}$  of  $s_{ch,i}$  with respect to  $s_{ch,standard}$  was calculated. In the first study a slight influence on the initial spacing was assumed and the value of 2.25 was changed by one tenth of  $\Delta s_{rel,i}$ . By this method the initial spacing covers a range from 2.168 mm to 2.639 mm in case of a hardness of 44 HRC. In the second study the full impact of the driving force was assumed and the initial spacing for the current data set was set to  $2.25 * \Delta s_{rel,i}$  which led to a range from 1.432 mm to 6.136 mm for the same hardness.

At first the results on 44 HRC will be discussed. Afterwards these results will be compared to the ones for 47 HRC and 50 HRC. In the first study the shape of the surface changes and the maximum number of load cycles drops by 170000 cycles which leads to a maximum increase of the factor 4.3. Furthermore the position of the maximum changes and so does the order of the data points at  $\alpha = 85\%$ . This can be explained by the influence of the residual stresses on the determination of  $s_{ch}$ . At  $\alpha = 85\%$  the residual



**Figure 4.23:** Characteristic crack spacings for variations of material parameters of a X38CrMoV5-1 hot work tool steel with a hardness of 44HRC.

stresses are very low especially at  $\lambda = 115\%$ . This data point marks the maximum in Figure 4.19 since the driving is lower than for the other parameter combinations and the spacing was assumed to be constant. If it is assumed that the driving force has an effect on the initial spacing as well, the situation changes. The low values of  $\sigma_{res}$  cause large spacings when a method based on driving forces is applied. This also means that the effect of crack shielding on crack propagation and service life is less pronounced. Thus, it is possible that smaller loads decrease service life. In the study in this section the stresses in the die affect both the initial spacing and crack growth. Due to crack shielding there is a buffer action between the two, but for each parameter combination usually either the influence on initial spacing or crack growth has a greater effect with regard to service life (this is an explanation for the complex shapes of the fit functions as will be discussed below).

The impact of driving force on crack growth, however, is the sole factor determining ser-



vice life. When talking about a buffer action in this context it is meant that the effect of the driving force on crack spacing determines how the driving force is shared between the cracks in a crack network. A large crack spacing means that the same driving force as in the case of a small spacing is shared between fewer cracks. Thus, each individual crack will grow faster. In case the overall driving force is small, however, the spacing has less impact because crack propagation will be slow anyway.

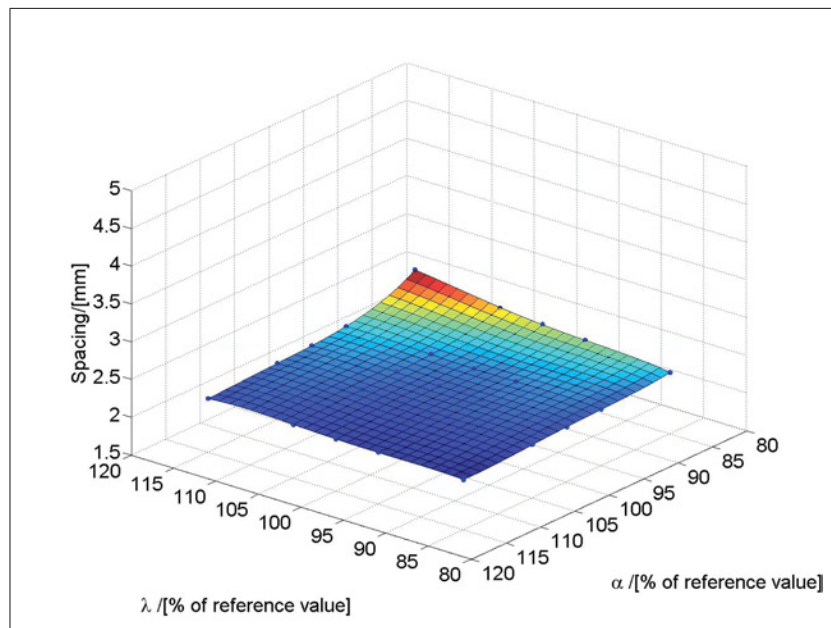
The results of the second study, where a larger influence of the driving force on the initial spacing was assumed, further suggest that in case the spacing is not determined by microstructure only, crack spacing and crack propagation buffer each other with respect to service life. Thus it is not possible to gain service life by a factor 6 as in the study on fixed model width. In the second study a maximum increase by a factor 1.4 is obtained. The buffer action between the effect of driving force on spacing and growth can be discussed when the edge on the left side of Figure 6.16 is regarded. The edge corresponds to 115% of  $\lambda_{standard}$ . With decreasing  $\alpha$  service life increases first and drops again for 95% and 85%  $\alpha_{standard}$ . The same edge in the study with fixed spacing increases steadily with lower values of  $\alpha$ . In the current study with variable spacing the crack distance rises along the edge until it reaches a maximum at 85%  $\alpha_{standard}$  and 115%  $\lambda_{standard}$ . Thus, the effect of crack shielding decays along this edge. Still service life increases at first, suggesting a predominant influence of the driving force on crack growth itself, and decreases when  $\alpha$  drops below 95% which is caused by the diminishing impact of crack shielding.

Additionally the shape of the surface described by the data in Figure 6.16 differs completely from the one in Figure 4.18. The global minimum and maximum are not at the same positions. There are two global minima and one local maximum of service life within the surface. If full impact of the driving force was applicable, a gain in service life by a factor of 1.2 could be obtained by decreasing  $\alpha$  by 5 percent and leaving  $\lambda$  as it is. This conclusion does not take into account other effects as mentioned in the introduction. More detailed studies based on advanced concepts are left to future work.

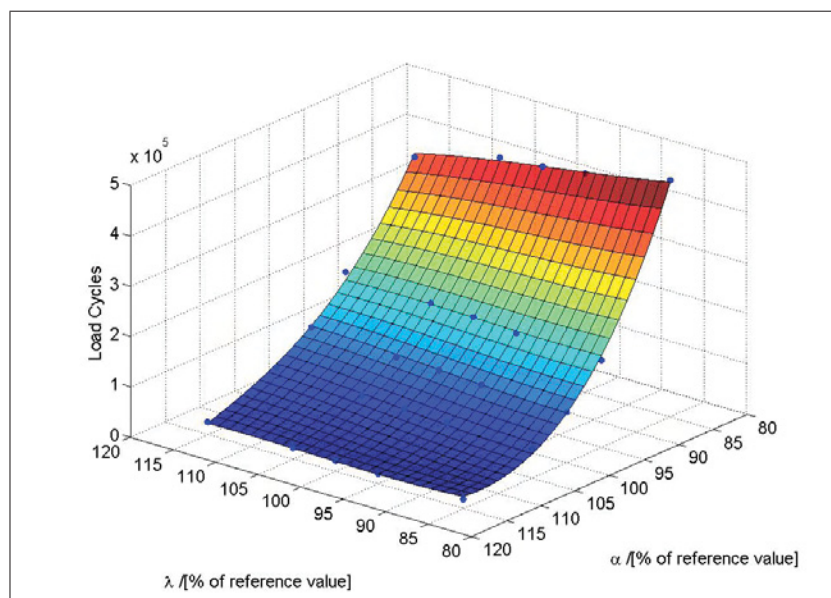
The studies with an assumed slight influence of the driving force on the crack spacing for 44 HRC and 47 HRC led to similar results. Service life for 47 HRC is somewhat shifted to higher values but the shape of the surface (Figures 4.25 and 4.27) stays the same. This can be explained by the similar distribution of residual stresses for these two hardnesses in the investigated parameter range. The variation of  $\sigma_{res}$  for 44 HRC is larger but this does not cause much difference in the results in this study. The shape of the surface for 50 HRC (Figure 4.29) is noticeable different, however. The hot work tool steel only shows limited plasticity at a hardness level of 50 HRC and thus develops less residual stresses. For a significant number of parameter combinations the  $\sigma_{res}$  is close to 0 which affects

the initial spacing significantly which in turn has an impact on service life (for data on residual stresses refer to the appendix). The drop in service life in Figure 4.25 is caused by the drop in residual stresses and the jump in initial spacing. The fit function slightly undershoots in this area.

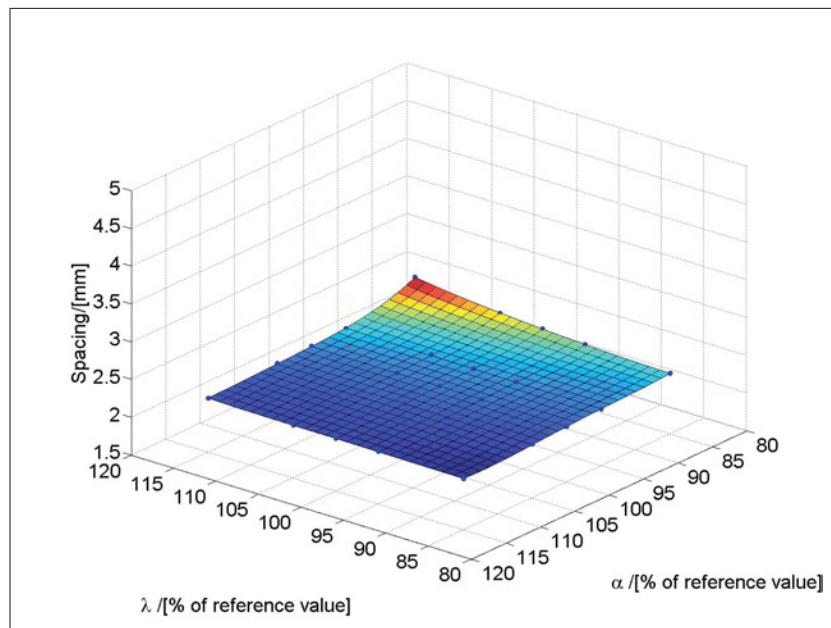
When a 100% proportionality between the characteristic and the real crack spacing is assumed, the shape of the data fit for 50 HRC (Figure 4.35) is similar to the one for constant spacing (Figure 4.21). In contrast to Figure 4.21 the surface rises in the right front corner, but the maximum is at the same position. The low residual stresses cause very wide spacings. Those, however, do not lower service life since driving forces are so low that even with little crack shielding propagation rates are low. Thus, the buffer action does not have much impact in this case. The corresponding surface for 47 HRC shown in Figure 4.33 is rather flat compared to Figure 4.31. This can be explained by the lower variation of  $\sigma_{res}$  for 47 HRC which did not have much effect in the study where a slight influence of driving force was assumed. It clearly affects the surface shapes when the differences in the spacing which depend on  $\sigma_{res}$  are more pronounced.



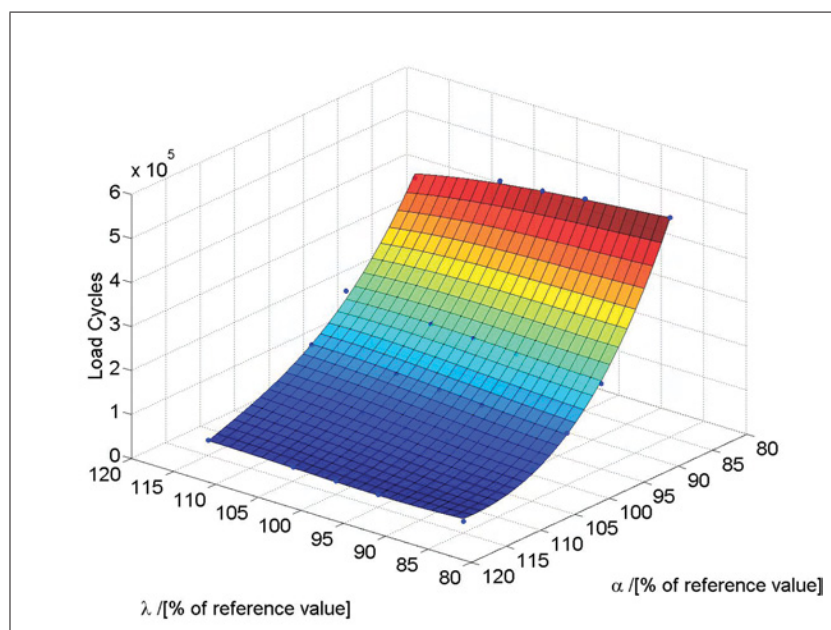
**Figure 4.24:** Initial crack spacing of a crack array based on the observed standard spacing 2.25 mm and a 10% influence of the driving force. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 44 HRC.



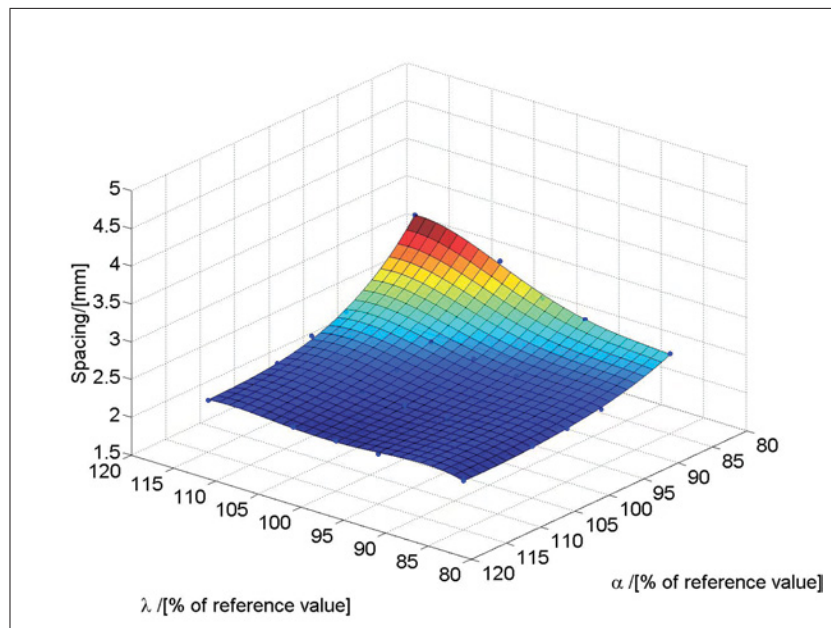
**Figure 4.25:** Number of load cycles necessary for a crack in the array to grow from 0.1 mm to 1.5 mm based on initial spacings depicted in Figure 4.24. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 44 HRC.



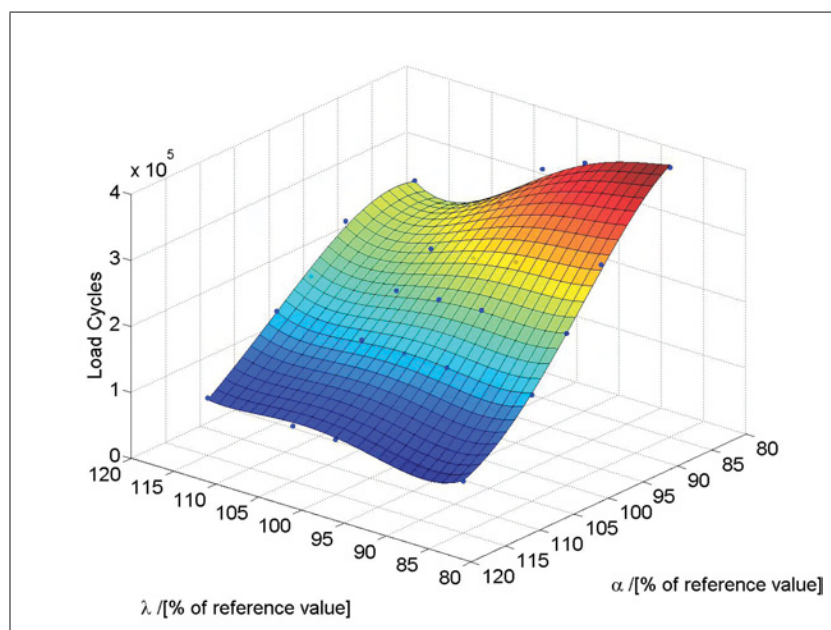
**Figure 4.26:** Initial crack spacing of a crack array based on the observed standard spacing 2.25 mm and a 10% influence of the driving force. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 47 HRC.



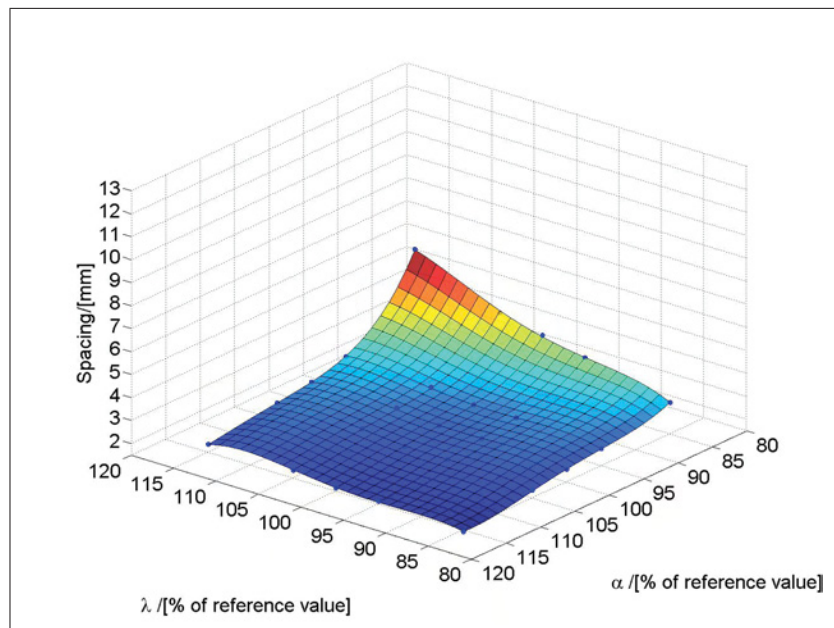
**Figure 4.27:** Number of load cycles necessary for a crack in the array to grow from 0.1 mm to 1.5 mm based on initial spacings depicted in Figure 4.26. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 47 HRC.



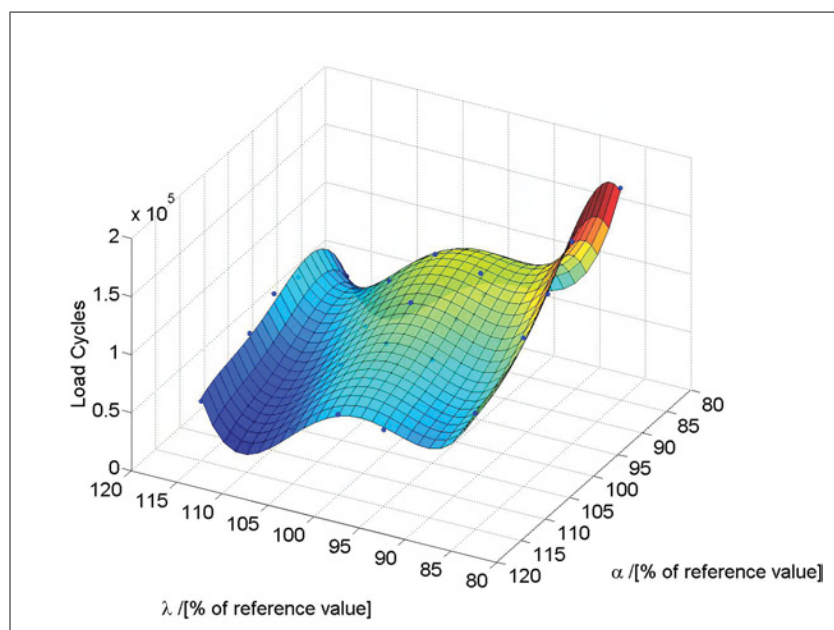
**Figure 4.28:** Initial crack spacing of a crack array based on the observed standard spacing 2.25 mm and a 10% influence of the driving force. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 50 HRC.



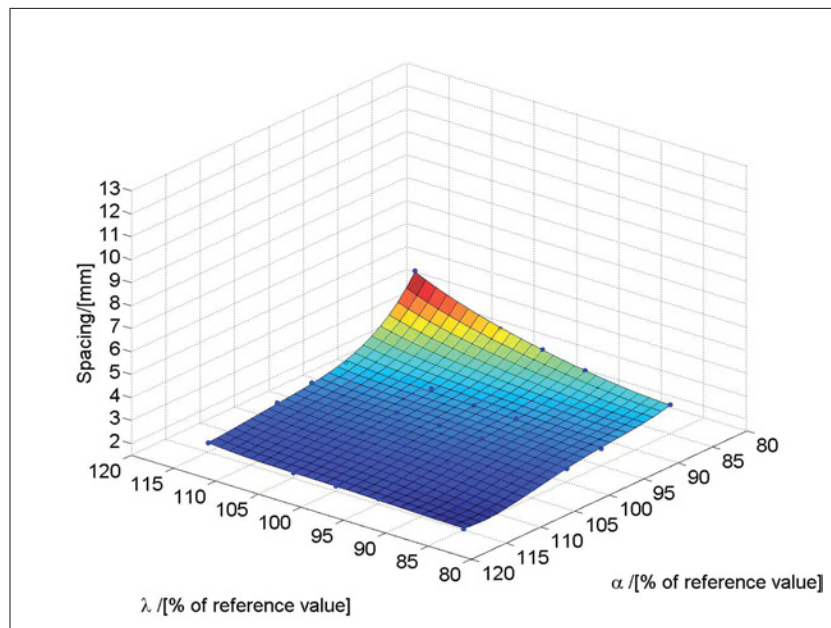
**Figure 4.29:** Number of load cycles necessary for a crack in the array to grow from 0.1 mm to 1.5 mm based on initial spacings depicted in Figure 4.28. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 50 HRC.



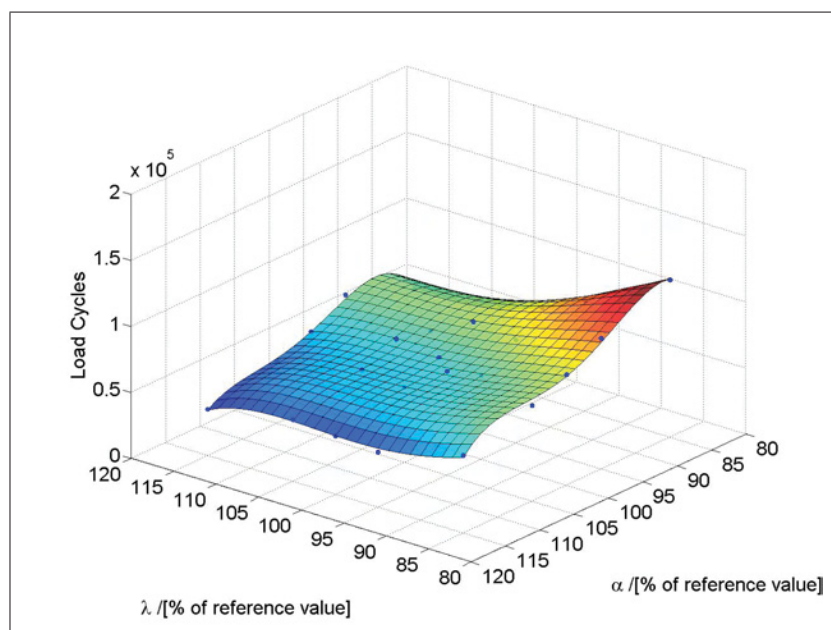
**Figure 4.30:** Initial crack spacing of a crack array based on the observed standard spacing 2.25 mm and a 100% influence of the driving force. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 44 HRC.



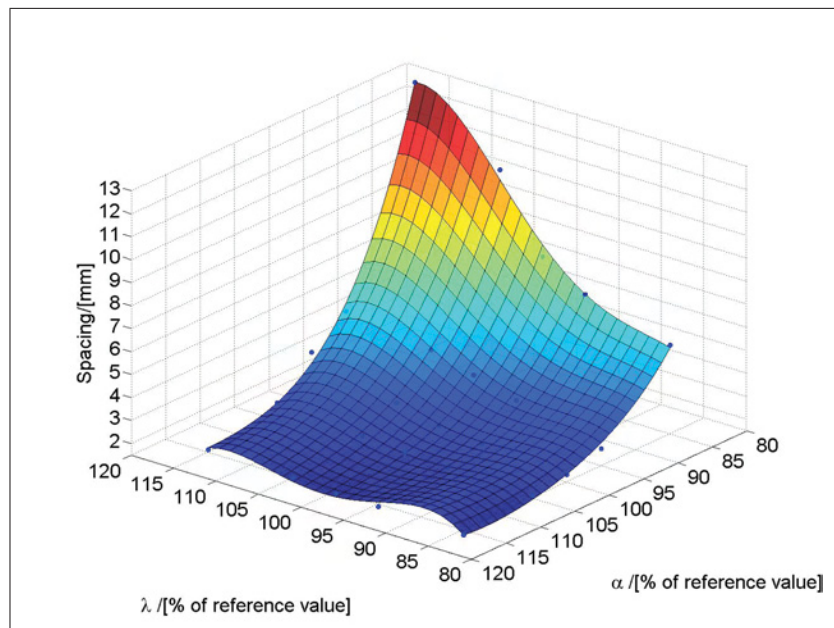
**Figure 4.31:** Number of load cycles necessary for a crack in the array to grow from 0.1 mm to 1.5 mm based on initial spacings depicted in Figure 4.30. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 44 HRC.



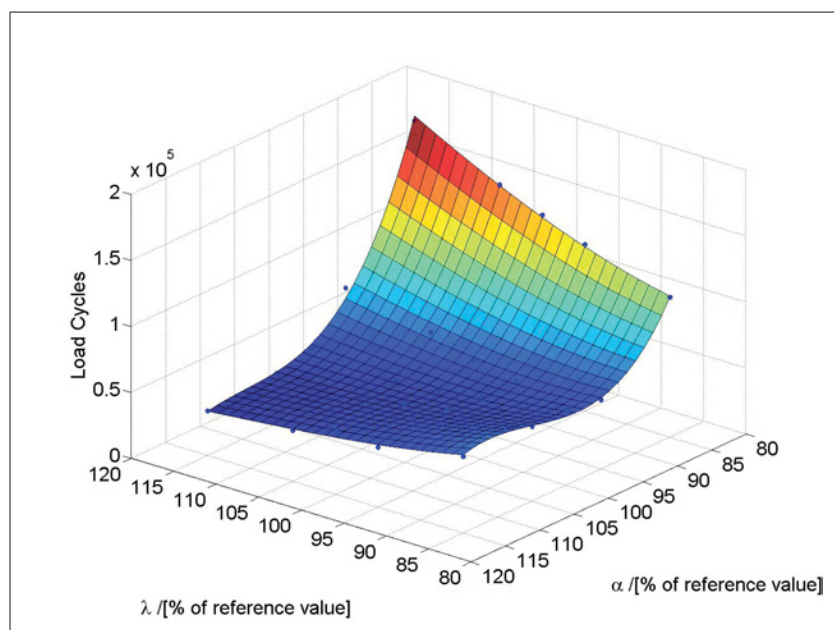
**Figure 4.32:** Initial crack spacing of a crack array based on the observed standard spacing 2.25 mm and a 100% influence of the driving force. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 47 HRC.



**Figure 4.33:** Number of load cycles necessary for a crack in the array to grow from 0.1 mm to 1.5 mm based on initial spacings depicted in Figure 4.32. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 47 HRC.



**Figure 4.34:** Initial crack spacing of a crack array based on the observed standard spacing 2.25 mm and a 100% influence of the driving force. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 50 HRC.



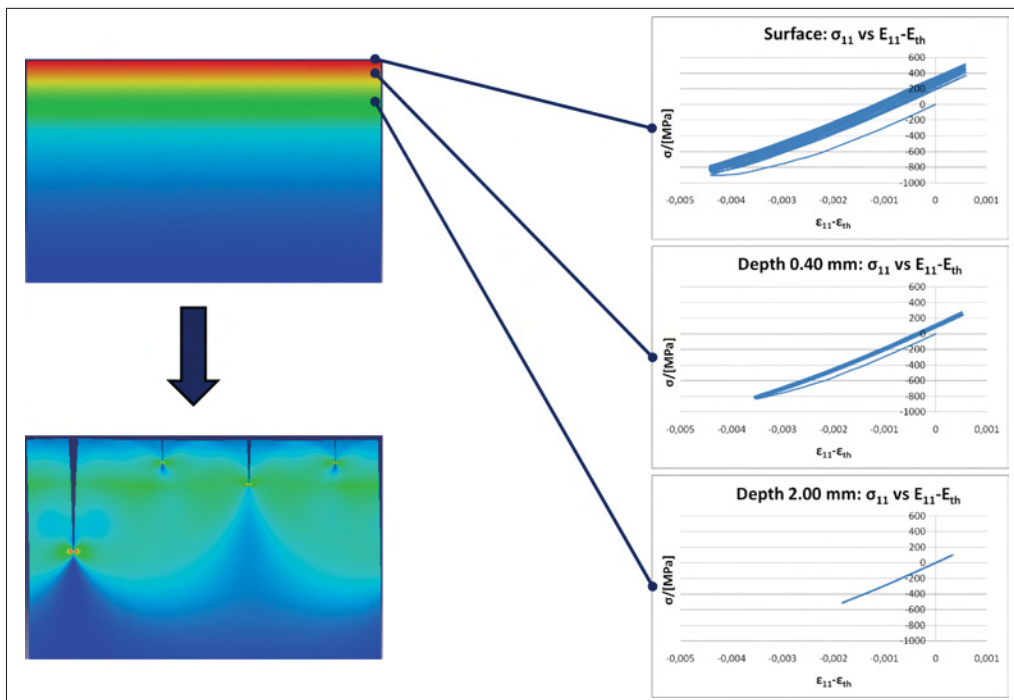
**Figure 4.35:** Number of load cycles necessary for a crack in the array to grow from 0.1 mm to 1.5 mm based on initial spacings depicted in Figure 4.34. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 50 HRC.



## 4.8 Limitations of the C-Model

The cyclic plastic zone lies, in case of standard material parameters, embedded within the top 0.5 millimeters of the die as shown in Figure 4.36. In this region neither LEFM, nor the J-Integral are precise concepts and a more suitable approach like the CTOD would be required. The CTOD is independent of contour integrals and therefore not subject to limitations from the stress and strain fields. This concept, however, would necessitate the change to an elastic-plastic model not applicable in the course of this study and is subject to future work. Furthermore its relation to Paris equation and crack growth would have to be investigated.

Since additional models for crack initiation are required and growth of small cracks follows a different behavior as shown in Figure 2.8, the present model cannot capture processes in the top 10-15  $\mu\text{m}$  of the material.



**Figure 4.36:** Transition from cyclic plastic to cyclic elastic zone.

# Chapter 5

## Discussions and Conclusions

The modeling path designed for this thesis is both built upon concepts which have already been used in other works and upon new approaches. This combination of old and new ideas allows the simulation of the evolution of crack arrays in a user friendly and automated manner. The time demand for the simulations shrinks from weeks to days and the models provide also new features like an automated evaluation of the results. In the present work the new models are used for the investigation of single crack growth, crack shielding and service life calculations.

In case of single crack propagation, where crack shielding is not involved, larger values of  $\lambda$  decrease  $\frac{da}{dN}$  in the surface region because the temperature gradient and thus the driving force for crack growth sinks. Due to the change of the temperature distribution the area where stresses change from tension to pressure is shifted to larger depths. This means that an increase of  $\lambda$  also increases the maximum crack length. Similar conclusions for the crack arrays are more difficult to obtain since crack shielding is of major importance. As shown in Figure 4.10 the influence of crack shielding takes effect over a very large distance. Its influence diminishes after 20 times the crack length. The influence of crack shielding on service life was shown to be significant. Small changes in the initial crack spacing of crack networks have an enormous impact. Figure 4.15 gives an impression how a few tenths of a millimeter in crack distance can cause changes in service life by several 10000 cycles.

So far there have been three factors identified which can influence the crack spacing. These are the microstructure, grinding grooves and the driving force which is related to the stresses in the die material. The first two have been shown to have significant importance, but the role of the stresses and the elastic strain energy in the dies after spraying with respect to the formation of the initial spacing of the crack arrays has not yet been

determined. The stresses and the strain energy depend on  $\lambda$  and  $\alpha$  which change the thermal loading situation if the process itself stays the same. Since only a limited number of real materials has been investigated no conclusions can be made about which effect determines crack spacing or if all of them have an influence. In this work estimates on service life are made which are both based on fixed initial spacings given by microstructure and surface treatments and variable spacings which are related to the driving force in order to investigate possible design rules.

The studies on service life for fixed spacings which were determined from real dies show that an increase in  $\lambda$  and a decrease in  $\alpha$  by 15% can lead to a large improvement of the service life of the dies by a factor of 5 or 6. On the one hand it is shown that a smaller amount of plasticity as in the case of a X38CrMoV5-1 hot work tool steel with a hardness of 50 HRC compared to the same type with 44 HRC, causes longer service life. This conclusion is dangerous because it does not consider that the material with 50 HRC is prone to single crack propagation which can cause early failure being the reason why lower hardnesses are preferred by die designers. On the other hand it is not considered that lower loads which result from a high thermal conductivity and low thermal expansion coefficients, may entail larger initial crack spacings.

The possible impact on service life is studied by relating the spacing of each material parameter combination of  $\lambda$  and  $\alpha$  to a characteristic minimum spacing which is determined by a criterion based on fracture mechanics. As the main result of these studies it turns out that there is a buffer action limiting the gain in service life. A decrease of the load which can be achieved by optimizing the material parameters lowers the driving force for crack growth. In this study it is taken into account that the driving force also affects the initial crack spacing which is essential with regard to service life. A low driving force leads to larger distances between the cracks thus reducing the shielding effect. This in turn entails larger driving forces for crack growth compared to the case of a fixed spacing. Altogether, it can be said that the gain in service life by optimization of material parameters is less than in the study with a fixed spacing. The shape of the resulting fit surfaces can significantly differ due to the fact that - despite the buffer action of a change of material parameters - either the effect on crack spacing or the direct effect on driving force for crack propagation dominates. This explains the different surface shapes in Figures 4.31 and 4.35 which correspond to studies on service life with variable initial crack spacing for a hardness of 44 HRC and 50 HRC, respectively. In case of 50 HRC residual stresses almost disappear for slightly decreased values of  $\alpha$  and slightly larger values for  $\lambda$ . This decreases the driving force significantly and results in large spacings. Despite a lower shielding effect crack propagation is still slow due to the driving force staying low even without shielding. Thus the direct effect of material parameters on driving force clearly

dominates. In case of 44 HRC there is more buffer action and thus the curve shape is different.

Since the combined impact of the microstructure, the surface treatment and the driving force on the initial crack spacing has not been investigated experimentally so far since it is almost impossible to produce and test the required amount of samples, the studies on service life are performed based on different scenarios. First, the material parameters are assumed not to change the observed crack spacing. The second and third scenarios are based on the assumption that the loading affects the spacing to some extent. The studies suggest that results for fixed and variable spacings can differ significantly. The possible gain in service life is lower if driving forces affect the spacing. Then the positions of minima and maxima change due to the effect of buffer actions.

The studies show that in any case a decrease in thermal expansion has a positive effect on service life. The question whether a large or small thermal conductivity is favorable can be answered only after further studies investigating the influence on crack spacing which determines life time to a great extent. Even if the stresses in the dies do not change spacing directly it is probable that they affect the number of eligible crack initiation sites in the microstructure. As discussed in Section 4.7 cracks have been found to initiate at different microstructural features depending on the amount of plastic strain. The fact that the latter depends on  $\lambda$  and  $\alpha$  gives rise to yet another mechanism that makes the spacing dependent on the loading.

The complex dependence of crack spacing on various factors can be simplified by purposefully engineering the spacing by means of appropriate surface treatments. If such treatments are found the effect of other influences like the microstructure is of less significance or none at all. This way crack spacing and crack propagation can be treated separately and the buffer action can be avoided which guarantees that a lower loading level will increase service life.

Once the discussed dependencies have been quantified experimentally the script-based modeling concept can be enhanced and reapplied for verification purposes. The methodology presented in this thesis proves versatile enough to accommodate virtually any kind of newly found phenomena.

# List of Figures

1.1	Strain based Wöhler curve illustrating the correlation of service life and hardness [2]. A larger hardness is favorable in case of small strains but has a negative effect or none at all if larger strain ranges prevail. . . . .	6
1.2	Steps of a pressure die casting cycle [4]. The heat flux vectors are indicated by arrows. . . . .	7
1.3	Sketch of a typical crack configuration and the boundary conditions used in the FE analysis [7]. . . . .	8
2.1	Load-displacement curves for increasing crack lengths and strain and stress control, respectively [13]. . . . .	14
2.2	Schematic driving force vs. R curve diagrams (a) flat R curve (b) rising R curve [12]. . . . .	15
2.3	Correction of the plastic zone size. The presence of plasticity makes the crack behave as if it were longer. Thus, the first order estimate $r_y$ on the plastic zone size has to be added to the actual crack length [12]. . . . .	18
2.4	Hinge model for CTOD estimation from three-point bend specimens [12]. . . . .	20
2.5	(a) Area of J dominated zone and (b) corresponding sections of $\sigma$ - $\epsilon$ diagram [15]. . . . .	22
2.6	HRR field and K dominated zone in small scale yielding [12]. . . . .	23
2.7	Overview of parameters used to characterize cyclic loading in terms of stress intensity [20]. . . . .	24
2.8	Regimes of stable fatigue crack propagation. Regime A: Near threshold fatigue, Regime B: Crack growth according to Paris equation, Regime C: Rapid crack growth. [14]. . . . .	26

2.9	Schematic of stress based Wöhler curve. The larger the stress level gets the more important becomes the crack propagation with regard to service life. At lower stress levels the impact of crack initiation is predominant [13].	27
2.10	One possible form of a Haigh Diagram. The curve found by Soderberg gives a conservative estimate for most engineering alloys. The Goodman relation gives very good results for brittle metals and the Gerber relation is appropriate for ductile metals and tensile mean stresses [23]. . . . .	28
2.11	Strain based Wöhler curve illustrating the findings of Basquin, Manson and Coffin [14]. . . . .	29
2.12	Typical ranges of fatigue threshold for some engineering alloys. Fe and Ni show a plateau which corresponds to an intrinsic threshold $\Delta K_{th}$ [14]. . . .	30
2.13	Schematic relation between the maximum stress intensity and the stress intensity range which illustrates the possibility to treat the onset of crack growth in both terms [14]. . . . .	31
2.14	Number of fatigue cycles to failure based on LEFM and small crack growth kinetics. The latter one gives nonconservative results for small initial sizes [14]. . . . .	32
2.15	Illustration of the relationship between applied stress and displacement with crack closure involved [14]. In the linear section between points A and B the crack flanks do not have contact. Between point B and point C contact is established. After point C contact is fully established. . . . .	37
2.16	Influence of R on the intrinsic threshold value $\Delta K_{th}$ . Below $R^*$ the apparent threshold for crack propagation is larger than the intrinsic value because of crack closure [12]. . . . .	37
2.17	Different mechanisms of crack closure: (a) plasticity-induced crack closure; (b) oxide-induced crack closure; (c) roughness-induced crack closure; (d) fluid-induced crack closure; (e) transformation-induced crack closure; (f) crack deflection; (g) crack-bridging by fibers; (h) crack-bridging by particles [14]. . . . .	38
2.18	Schematic of plasticity induced crack closure under plane stress conditions. Extra material at the crack flanks originates from plastic out of plane flow [28]. . . . .	39

2.19	On the left hand side: Volume transfer to the crack tip by rotation of volume elements in the wake during crack growth under plane strain conditions; At the center: Monotonic and cyclic plastic zone; On the right hand side: Remaining ligament [28]. . . . .	40
2.20	Interaction of two identical coplanar through thickness cracks in an infinite plate [12]. . . . .	41
2.21	Interaction of two identical parallel through thickness cracks in an infinite plate [12]. . . . .	42
2.22	Temperature distribution due to convection, redrawn from [29]. . . . .	43
2.23	Two critical states during crack propagation with increasing $\delta$ , redrawn from [29]. . . . .	45
2.24	Determination of minimum spacing, redrawn from [31]. The expression $\frac{L}{L_c}^2$ is proportional to time. After the cracks start to initiate the optimum spacing decreases. The global minimum determines the optimum spacing observed in the material. . . . .	45
2.25	Period doubling during crack propagation in a growing layer, redrawn from [31]. The black line is valid for a crack array with the optimum initial spacing which is illustrated in Figure 2.24. The dashed line corresponds to crack configurations with the current optimum spacing in each time increment. . . . .	46
3.1	Flow chart of the modeling path. . . . .	49
3.2	Overview of data transfer to the main script. . . . .	54
3.3	Abaqus schematic, modified from [34]. The orange part illustrates the design of the Abaqus preprocessor (Abaqus CAE). The red section describes the solvers provided by the Abaqus FE package. The blue section refers to postprocessor which is integrated in Abaqus CAE and the two types of output, i.e. field and history output. . . . .	56
3.4	Relation between Control Script and C-Model. When the script is started user defined modules are loaded. These contain data required for the application of a built-in criterion and functions needed during the execution of the script. . . . .	57

3.5	Progression of $K$ for all cracks in an array with an optimum minimum initial spacing of $230 \mu m$ . The crack configurations where cracks stop are designated as instabilities. The stress intensities of cracks which have stopped drop significantly upon the propagation of the other cracks. . . . .	59
3.6	Envelope of all increments of a solidification step. This envelope corresponds to the distribution of $\sigma_{min}$ . . . . .	60
3.7	Crack propagation curves for a X38CrMoV5-1 hot work tool steel with a hardness of 44 HRC for various stress ratios [7]. . . . .	62
3.8	The crack propagation curves are shifted in order to obtain a $\Delta K_{eff}$ curve which is compared to Paris relation determined in [7]. . . . .	64
3.9	Data obtained from the shift of crack propagation curves (Figure 3.8) is used to define $\Delta K_{eff}/K_{max}$ as a function of $R$ . . . . .	64
3.10	Progression of $\Delta K_{eff}$ . For the integration of the Paris equation the progression is approximated by linear sections. . . . .	67
3.11	In order to find the characteristic minimum spacing of a parameter set the width of a model containing a $100 \mu m$ long crack is increased until $\Delta K_{eff}$ reaches the threshold value $\Delta K_{th}$ . . . . .	69
4.1	Temperature distributions for various parameter variations. Larger values of $\lambda$ cause smaller temperature gradients but also shift the point where stresses change from tension to pressure to larger depths, i.e. the intersection of temperature curves with the background temperature ( $120^\circ C$ ). . . . .	72
4.2	Distribution of residual stresses after 100 casting cycles in X38CrMoV5-1 hot work tool steels for three different heat treatments as initial states before cyclic loading. . . . .	74
4.3	Distribution of residual stresses after 100 casting cycles in a X38CrMoV5-1 hot work tool steel with a hardness of 44HRC for different material parameters. . . . .	74
4.4	Overview of residual stresses at the surface of a X38CrMoV5-1 hot work tool steel with a hardness of 44HRC after 100 casting cycles for different material parameters. . . . .	75
4.5	Polynomial fit of data shown in Figure 4.4. . . . .	76
4.6	Distributions of $\sigma_{min}$ for different material parameters. . . . .	77



---

4.7	Distributions of $\sigma_{max}$ for various initial configurations. . . . .	78
4.8	Two characteristic models. a: $C - Model_{\sigma_{max}}$ for initial spacing 1 mm; b: $C - Model_{\sigma_{max}}$ for initial spacing 2 mm . . . . .	79
4.9	Stress ratios for different datasets. . . . .	80
4.10	The influence of crack shielding covers a distance of about 20 times the crack length. . . . .	81
4.11	Data on single crack propagation. . . . .	83
4.12	Data on a crack array with with the closest possible initial spacing. . . . .	84
4.13	Scheme of modeling the development of a crack array. The contours show the normal stress distribution in 1-direction. . . . .	86
4.14	Crack arrays at two different positions of a die received from industry after service life was reached. . . . .	87
4.15	Progression of load cycles with increasing crack length for various initial spacings. This diagram shows that a small change of the initial spacing has a significant effect on service life. . . . .	88
4.16	Load cycles per increment for various initial spacings. . . . .	88
4.17	Propagation rates for various initial spacings. . . . .	89
4.18	Study on service life for a X38CrMoV5-1 hot work tool steel with a hardness of 44 HRC. The plot shows the number of load cycles until a crack length of 1.5 mm is reached. The initial crack spacing is 2.25 mm for all parameter combinations. . . . .	90
4.19	Polynomial fit of data shown in Figure 4.18 for 44 HRC. The axes were turned around for the sake of visibility. . . . .	92
4.20	Polynomial fit of data on service life for 47HRC. The initial crack spacing is 2.25 mm for all parameter combinations. . . . .	93
4.21	Polynomial fit of data on service life for 50 HRC. The initial crack spacing is 2.25 mm for all parameter combinations. . . . .	94
4.22	The influence of microstructure on initial crack spacing. . . . .	97
4.23	Characteristic crack spacings for variations of material parameters of a X38CrMoV5-1 hot work tool steel with a hardness of 44HRC. . . . .	99

---

4.24	Initial crack spacing of a crack array based on the observed standard spacing 2.25 mm and a 10% influence of the driving force. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 44 HRC. . . . .	102
4.25	Number of load cycles necessary for a crack in the array to grow from 0.1 mm to 1.5 mm based on initial spacings depicted in Figure 4.24. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 44 HRC. . . . .	102
4.26	Initial crack spacing of a crack array based on the observed standard spacing 2.25 mm and a 10% influence of the driving force. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 47 HRC. . . . .	103
4.27	Number of load cycles necessary for a crack in the array to grow from 0.1 mm to 1.5 mm based on initial spacings depicted in Figure 4.26. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 47 HRC. . . . .	103
4.28	Initial crack spacing of a crack array based on the observed standard spacing 2.25 mm and a 10% influence of the driving force. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 50 HRC. . . . .	104
4.29	Number of load cycles necessary for a crack in the array to grow from 0.1 mm to 1.5 mm based on initial spacings depicted in Figure 4.28. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 50 HRC. . . . .	104
4.30	Initial crack spacing of a crack array based on the observed standard spacing 2.25 mm and a 100% influence of the driving force. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 44 HRC. . . . .	105
4.31	Number of load cycles necessary for a crack in the array to grow from 0.1 mm to 1.5 mm based on initial spacings depicted in Figure 4.30. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 44 HRC. . . . .	105
4.32	Initial crack spacing of a crack array based on the observed standard spacing 2.25 mm and a 100% influence of the driving force. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 47 HRC. . . . .	106
4.33	Number of load cycles necessary for a crack in the array to grow from 0.1 mm to 1.5 mm based on initial spacings depicted in Figure 4.32. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 47 HRC. . . . .	106
4.34	Initial crack spacing of a crack array based on the observed standard spacing 2.25 mm and a 100% influence of the driving force. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 50 HRC. . . . .	107

4.35	Number of load cycles necessary for a crack in the array to grow from 0.1 mm to 1.5 mm based on initial spacings depicted in Figure 4.34. Data for a X38CrMoV5-1 hot work tool steel with a hardness of 50 HRC. . . . .	107
4.36	Transition from cyclic plastic to cyclic elastic zone. . . . .	108
6.1	Overview of residual stresses at the surface of a X38CrMoV5-1 hot work tool steel with a hardness of 47HRC for different material parameters. . . .	125
6.2	Polynomial fit of data shown in Figure 6.1. . . . .	126
6.3	Overview of residual stresses at the surface of a X38CrMoV5-1 hot work tool steel with a hardness of 50HRC for different material parameters. . . .	127
6.4	Polynomial fit of data shown in Figure 6.3. . . . .	128
6.5	X38CrMoV5-1 44 HRC: Load cycles until a crack length of 1.5 mm is reached in dependence of $\alpha$ for fixed values of $\lambda$ . . . . .	129
6.6	X38CrMoV5-1 44 HRC: Load cycles until a crack length of 1.5 mm is reached in dependence of $\lambda$ for fixed values of $\alpha$ . . . . .	129
6.7	X38CrMoV5-1 47 HRC: Load cycles until a crack length of 1.5 mm is reached.	130
6.8	X38CrMoV5-1 47 HRC: Load cycles until a crack length of 1.5 mm is reached in dependence of $\alpha$ for fixed values of $\lambda$ . . . . .	131
6.9	X38CrMoV5-1 47 HRC: Load cycles until a crack length of 1.5 mm is reached in dependence of $\lambda$ for fixed values of $\alpha$ . . . . .	131
6.10	X38CrMoV5-1 50 HRC: Load cycles until a crack length of 1.5 mm is reached.	132
6.11	X38CrMoV5-1 50 HRC: Load cycles until a crack length of 1.5 mm is reached in dependence of $\alpha$ for fixed values of $\lambda$ . . . . .	133
6.12	X38CrMoV5-1 50 HRC: Load cycles until a crack length of 1.5 mm is reached in dependence of $\lambda$ for fixed values of $\alpha$ . . . . .	133
6.13	Load cycles until a crack length of 1.5 mm is reached including variable crack spacing based on the observed standard spacing 2.25 mm and a 10% influence of the driving force. Data on a X38CrMoV5-1 hot work tool steel with a hardness of 44 HRC. . . . .	134
6.14	Load cycles until a crack length of 1.5 mm is reached based on the observed standard spacing 2.25 mm and a 10% influence of the driving force in dependence of $\alpha$ for fixed values of $\lambda$ . . . . .	135

- 
- 6.15 Load cycles until a crack length of 1.5 mm is reached based on the observed standard spacing 2.25 mm and a 10% influence of the driving force in dependence of  $\lambda$  for fixed values of  $\alpha$ . . . . . 135
- 6.16 Load cycles until a crack length of 1.5 mm is reached including variable crack spacing based on the observed standard spacing 2.25 mm and a 100% influence of the driving force. Data on a X38CrMoV5-1 hot work tool steel with a hardness of 44 HRC. . . . . 136
- 6.17 Load cycles until a crack length of 1.5 mm is reached based on the observed standard spacing 2.25 mm and a 100% influence of the driving force in dependence of  $\alpha$  for fixed values of  $\lambda$ . . . . . 137
- 6.18 Load cycles until a crack length of 1.5 mm is reached based on the observed standard spacing 2.25 mm and a 100% influence of the driving force in dependence of  $\lambda$  for fixed values of  $\alpha$ . . . . . 137

# List of Tables

3.1	Important thermo-physical material parameters of a X38CrMoV5-1 [7]. . .	51
3.2	Variations of $\lambda$ , $\alpha$ and hardness used for the studies in the present work. .	52
3.3	Polynomials used for the data fit required in order to relate $K_{max}$ with $\Delta K_{eff}$ . The polynomials are depicted in Figure 3.9. . . . .	63
3.4	Format of the main output file. . . . .	69
4.1	Single crack propagation for three different values of $\lambda$ . An increase in thermal conductivity causes lower propagation rates at the surface but larger maximum depths. . . . .	72
4.2	Variations of $\lambda$ , $\alpha$ and hardness used in the core study on a X38CrMoV5-1 hot work tool steel. . . . .	90

# Bibliography

- [1] H. Berns. Beispiele zur Schädigung von Warmarbeitwerkzeugen. *Zeitschrift für Werkstoffe, Wärmebehandlung, Fertigung*, 59:379–387, 2004.
- [2] R. Ebner, S. Marsoner, I. Siller, W. Ecker. Thermal fatigue behaviour of hot work tool steels - heat check nucleation and growth. In *Tooling materials and their applications from research to market*, 2006.
- [3] A. Hoffman, I. Siller, R. Ebner. Werkstoffe für Problembereiche in Gusskokillen. *Giesserei Rundschau*, 50:117–122, 2003.
- [4] W. Ecker. *Modeling of the thermo-mechanical behavior of pressure casting dies*. PhD thesis, Montanuniversität Leoben, 2008.
- [5] W. Ecker, T. Antretter, R. Ebner. Calibration and validation of an elasto-viscoplastic material model for a hot work tool steel used in pressure die casting dies. *Key Engineering Materials*, 345-346:685–688, 2007.
- [6] W. Ecker, T. Antretter, R. Ebner, E. Kaschnitz, S. Marsoner, M. Leindl, H. Köberl. The cyclic behavior of a hot work tool steel used in pressure die casting. In *1. Leobener Betriebsfestigkeitstage*, 2006.
- [7] M. Leindl. *Untersuchung des Risswachstums in einem Druckgießwerkzeug mit Hilfe der Finite-Elemente-Methode*. PhD thesis, Montanuniversität Leoben, 2008.
- [8] X. Huang, T. Moan. Improved modeling of the effect of R-Ratio on crack growth rate. *International Journal of Fatigue*, 29:591–602, 2007.
- [9] W. Ecker, M. Leindl, T. Antretter, R. Ebner, I. Siller, S. Nissle. Effect of material properties on the thermo-mechanical loading and damage of pressure casting dies – a numerical study. In *Tool Steels – Deciding Factor in Worldwide Production. Proceedings of the 8th International Tooling Conference. Aachen*, 2009.

- 
- [10] M. Leindl, W. Ecker, R. Ebner, T. Antretter. Numerische und experimentelle Untersuchung von Temperaturwechselrissnetzwerken. In *Proceedings of the 2nd Fatigue Symposium Leoben*, 2008.
- [11] *www.simulia.com*.
- [12] T. L. Anderson. *Fracture Mechanics, Fundamentals and Applications*. CRC Press Inc, 2005.
- [13] R. W. Hertzberg. *Deformation and Fracture Mechanics of Engineering Materials*. Wiley, 1996.
- [14] S. Suresh. *Fatigue of materials*. Cambridge University Press, 1998.
- [15] D. Gross, T. Seelig. *Bruchmechanik, Mit einer Einführung in die Mikromechanik*. Springer, 2007.
- [16] G. R. Irwin. Onset of fast crack propagation in high strength steel and aluminum alloys. *Sagamore Research Conference Proceedings*, 2:289–305, 1956.
- [17] R. Pippan. Lecture notes on "Ausgewählte Kapitel der Festkörpermechanik", Montanuniversität Leoben, 2008/09.
- [18] F. O. Riemelmoser, R. Pippan. Consideration of the mechanical behaviour of small fatigue cracks. *International Journal of Fracture*, 118:251–270, 2002.
- [19] J. R. Rice. A path independent integral and the approximate analysis of strain concentration by notches and cracks. *Journal of Applied Mechanics*, 35:379–386, 1968.
- [20] J. C. Newman, W. Elber, editor. *Mechanics of Fatigue Crack Closure*. American Society for Testing and Materials, 1988.
- [21] R. Jones, B. Chen, S. Pitt. Similitude: Fatigue cracking in steels. *Theoretical and Applied Fracture Mechanics*, 48:161–168, 2007.
- [22] S. Pitt R. Jones, L. Molent. Similitude and the paris crack growth law. *International Journal of Fatigue*, 30:1873–1880, 2008.
- [23] L. Lawson, E. Y. Chen, M. Meshii. Near-threshold fatigue: a review. *International Journal of Fatigue*, 21:15–34, 1999.

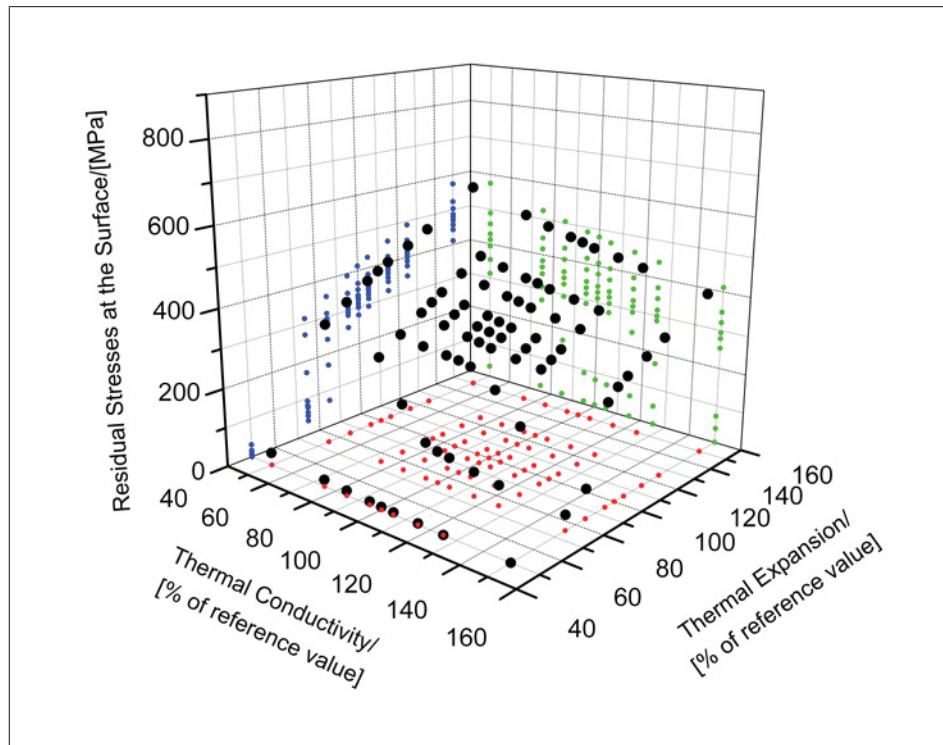
- [24] R. A. Schmidt, P. C. Paris. Threshold for fatigue crack propagation and the effects of load ratio and frequency. *Progress in Flaw Growth and Fracture Toughness Testing, ASTM STP 536*, pages 79–94, 1973.
- [25] A. K. Vasudevan, K. Sadananda. Classification of fatigue crack growth behavior. *Metallurgical and Materials Transactions A*, 26A:1221–1234, 1995.
- [26] C. R. Krenn, J. W. Morris Jr. The compatibility of crack closure and  $K_{max}$  dependent models of fatigue crack growth. *International Journal of Fatigue*, 21:147–155, 1999.
- [27] F. O. Riemelmoser, R. Pippan. Mechanical reasons for plasticity-induced crack closure under plane strain conditions. *Fatigue & Fracture of Engineering Materials & Structures*, 21:1425–1433, 1998.
- [28] R. Pippan, F. O. Riemelmoser. Visualization of the plasticity-induced crack closure under plain strain conditions. *Engineering Fracture Mechanics*, 60:315–322, 1998.
- [29] S. Nemat-Nasser, L. M. Keer, K.S. Parihar. Unstable growth of thermally induced interacting cracks in brittle solids. *International Journal of Solids and Structures*, 14:409–430, 1978.
- [30] H. Haddar, A. Fissolo, V. Maillot. Thermal fatigue crack networks: an computational study. *International Journal of Solids and Structures*, 42:771–788, 2005.
- [31] D. R. Jenkins. Determination of crack spacing and penetration due to shrinkage of a solidifying layer. *International Journal of Solids and Structures*, 46:1078–1084, 2009.
- [32] [www.nwnumerics.com](http://www.nwnumerics.com).
- [33] <http://docs.python.org>.
- [34] *Abaqus V6.8 Scripting User's Manual*.
- [35] *Abaqus V6.8 Scripting Reference Manual*.
- [36] G. Jesner. *Ermüdungsverhalten von hochfesten PM-Werkzeugstählen für die Kalt-massivumformung*. PhD thesis, Montanuniversität Leoben, 2007.
- [37] Y. Murakami, editor. *Stress Intensity Factors Handbook*. Pergamon Press, 1987.
- [38] R. Ebner. Lecture notes on "Spezielle Werkstoffprüfung", Montanuniversität Leoben, 2008/09.



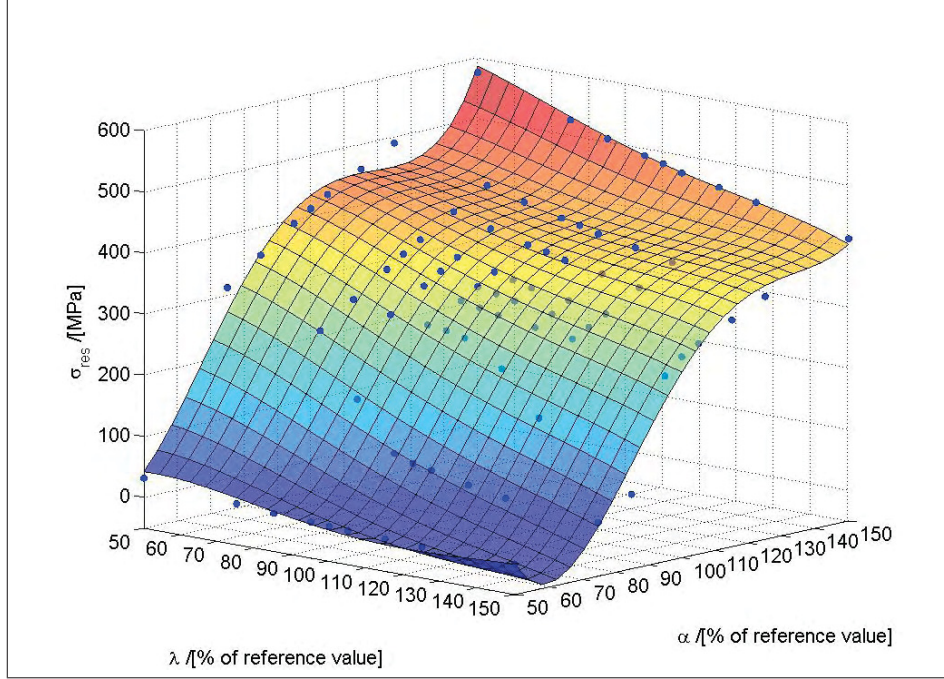
# Chapter 6

# Appendix

## Residual Stresses for 47 HRC and 50 HRC



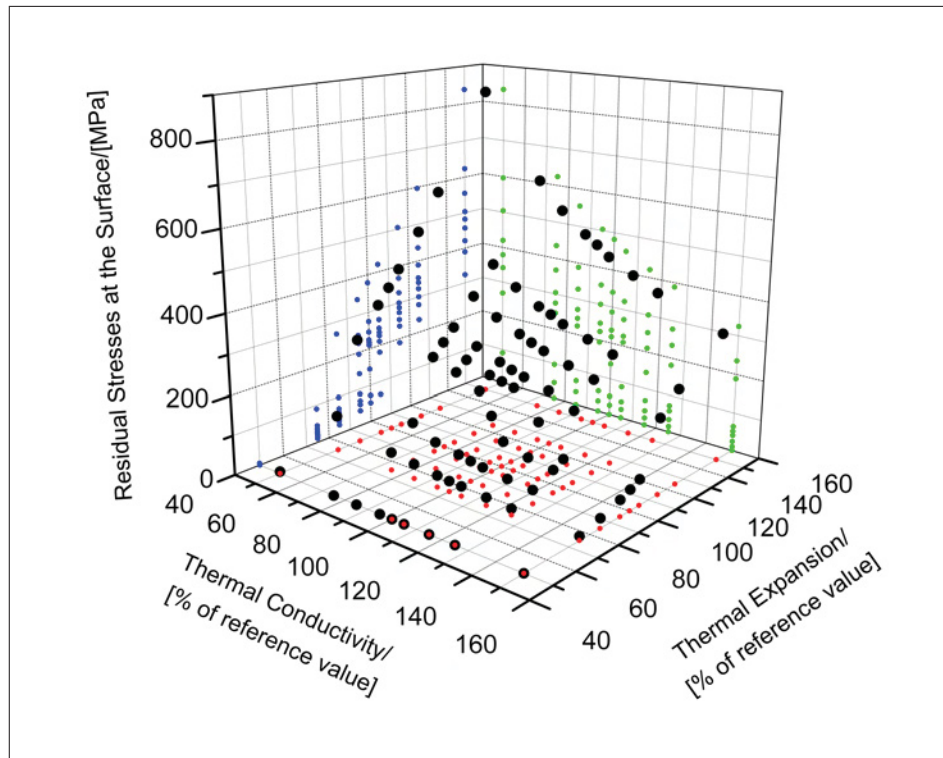
**Figure 6.1:** Overview of residual stresses at the surface of a X38CrMoV5-1 hot work tool steel with a hardness of 47HRC for different material parameters.



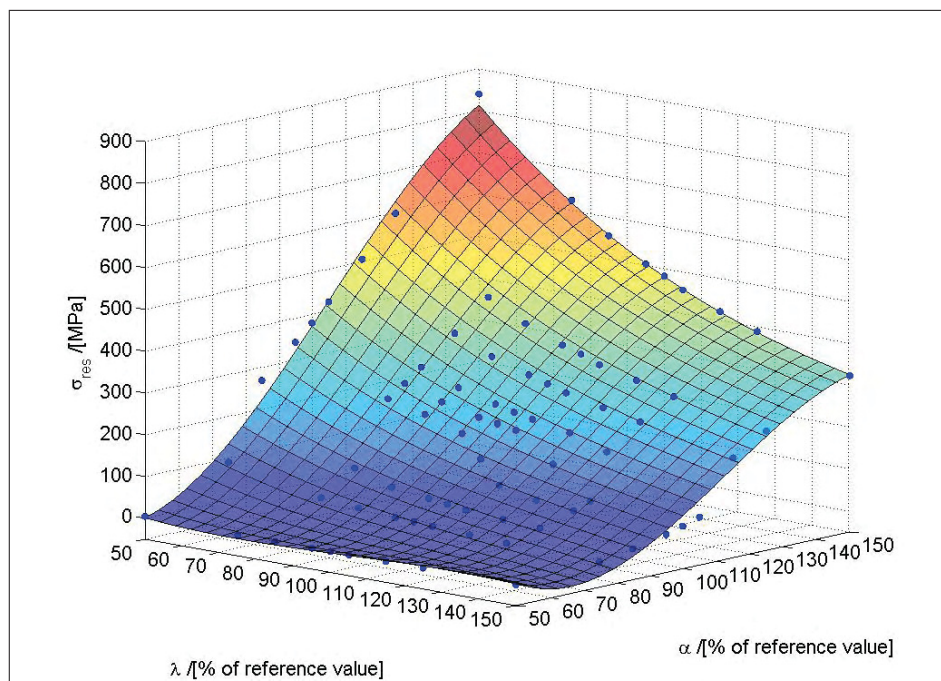
**Figure 6.2:** Polynomial fit of data shown in Figure 6.1.

$$\sigma_{res} = a_1\lambda^4 + a_2\alpha^4 + a_3\lambda^3\alpha + a_4\lambda\alpha^3 + a_5\lambda^2\alpha^2 + a_6\lambda^3 + a_7\alpha^3 + a_8\lambda^2\alpha + a_9\lambda\alpha^2 + a_{10}\lambda^2 + a_{11}\alpha^2 + a_{12}\lambda\alpha + a_{13}\lambda + a_{14}\alpha + a_{15} \quad (6.1)$$

$$\vec{a} = \begin{pmatrix} -0.00000340592094710405 \\ 0.0000360028726405065 \\ -0.00000197350669864322 \\ -0.0000242447570293971 \\ 0.000000963336380791848 \\ 0.00159374435292111 \\ -0.0121799129058228 \\ 0.000377413237764309 \\ 0.00735407750205425 \\ -0.248866479570291 \\ 1.34906126542654 \\ -0.746742898301332 \\ 35.5107562437758 \\ -46.5373471886697 \\ 21.114459154151 \end{pmatrix} \quad (6.2)$$



**Figure 6.3:** Overview of residual stresses at the surface of a X38CrMoV5-1 hot work tool steel with a hardness of 50HRC for different material parameters.

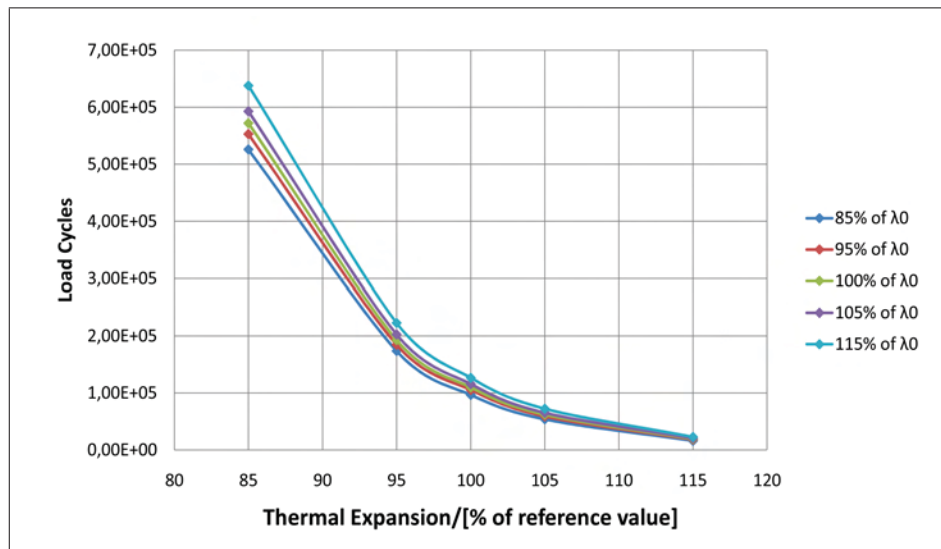


**Figure 6.4:** Polynomial fit of data shown in Figure 6.3.

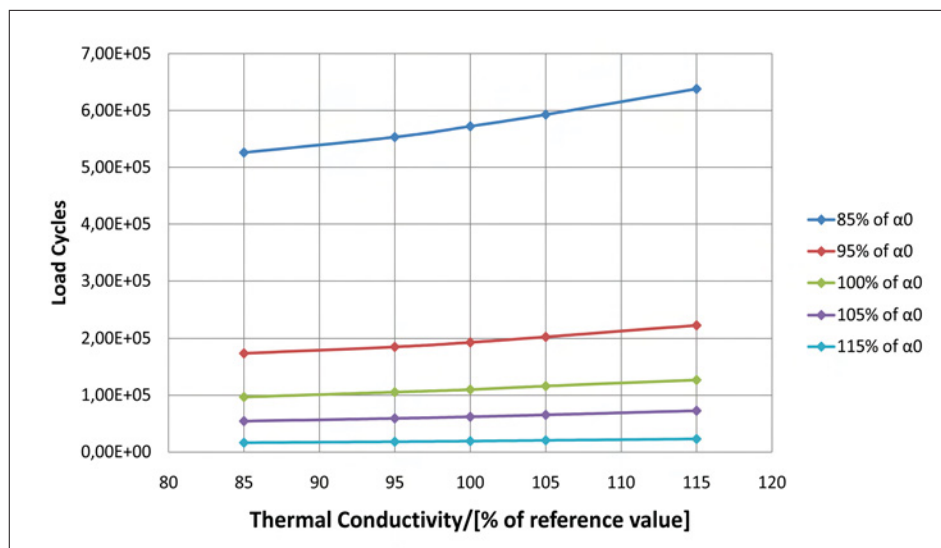
$$\sigma_{res} = a_1\lambda^3 + a_2\alpha^3 + a_3\lambda^2\alpha + a_4\lambda\alpha^2 + a_5\lambda^2 + a_6\alpha^2 + a_7\lambda\alpha + a_8\lambda + a_9\alpha + a_{10}$$

$$\vec{a} = \begin{pmatrix} -0.000134499631382486 \\ -0.000916214607330973 \\ 0.000326785519921527 \\ 0.000209427345161707 \\ 0.026648612381805 \\ 0.278753415998307 \\ -0.156170759758785 \\ 3.09111672214502 \\ -12.9724846111266 \\ 186.569095401594 \end{pmatrix} \quad (6.3)$$

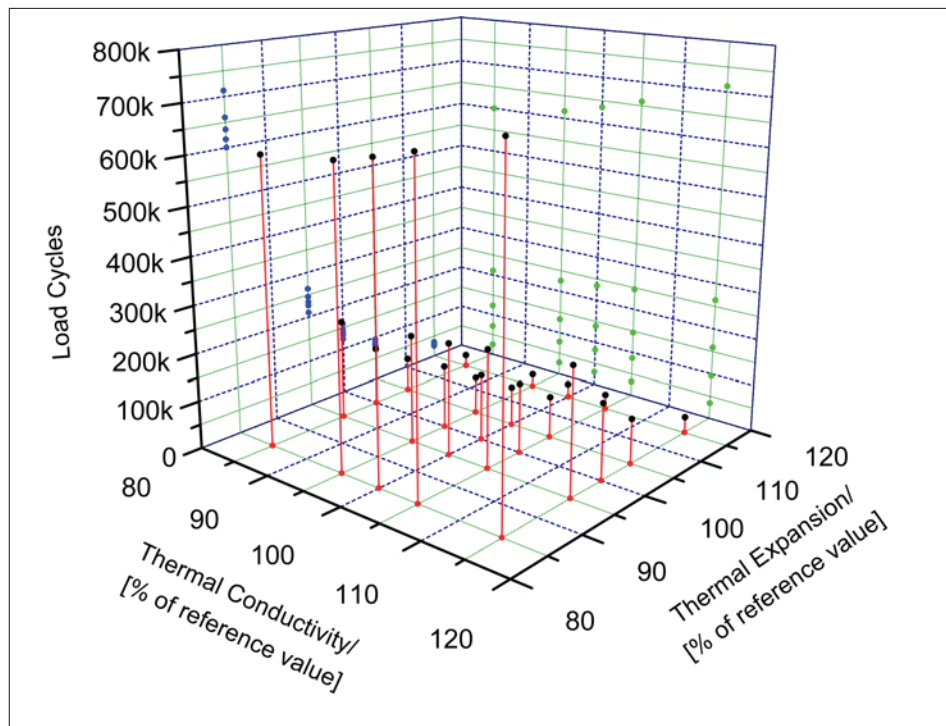
### Influence of Material Parameters on the Evolution of Crack Arrays in Case of Constant Initial Spacing-Detailed Results



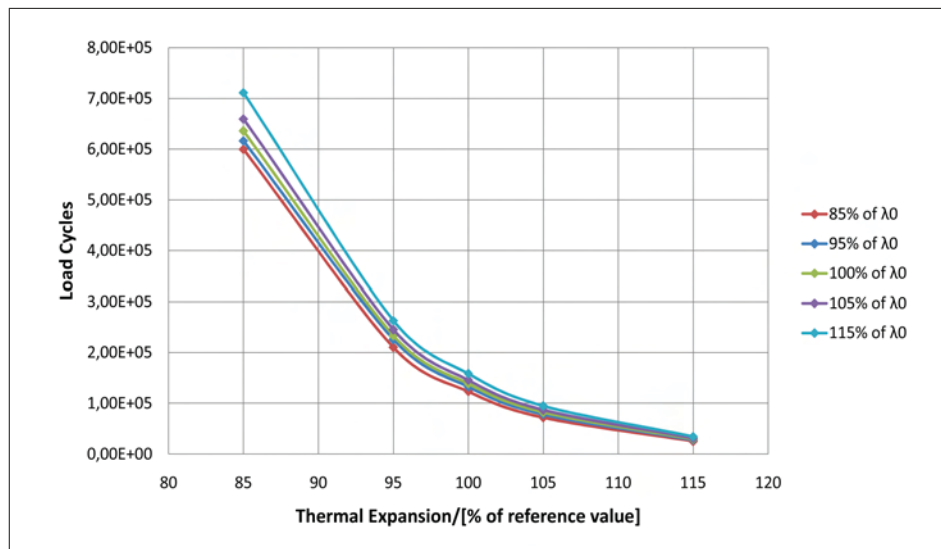
**Figure 6.5:** X38CrMoV5-1 44 HRC: Load cycles until a crack length of 1.5 mm is reached in dependence of  $\alpha$  for fixed values of  $\lambda$ .



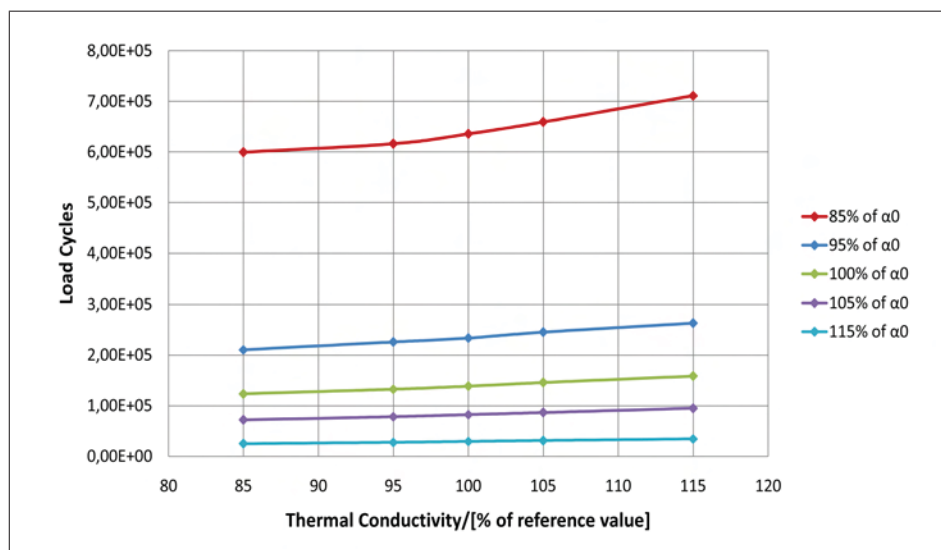
**Figure 6.6:** X38CrMoV5-1 44 HRC: Load cycles until a crack length of 1.5 mm is reached in dependence of  $\lambda$  for fixed values of  $\alpha$ .



**Figure 6.7:** X38CrMoV5-1 47 HRC: Load cycles until a crack length of 1.5 mm is reached.

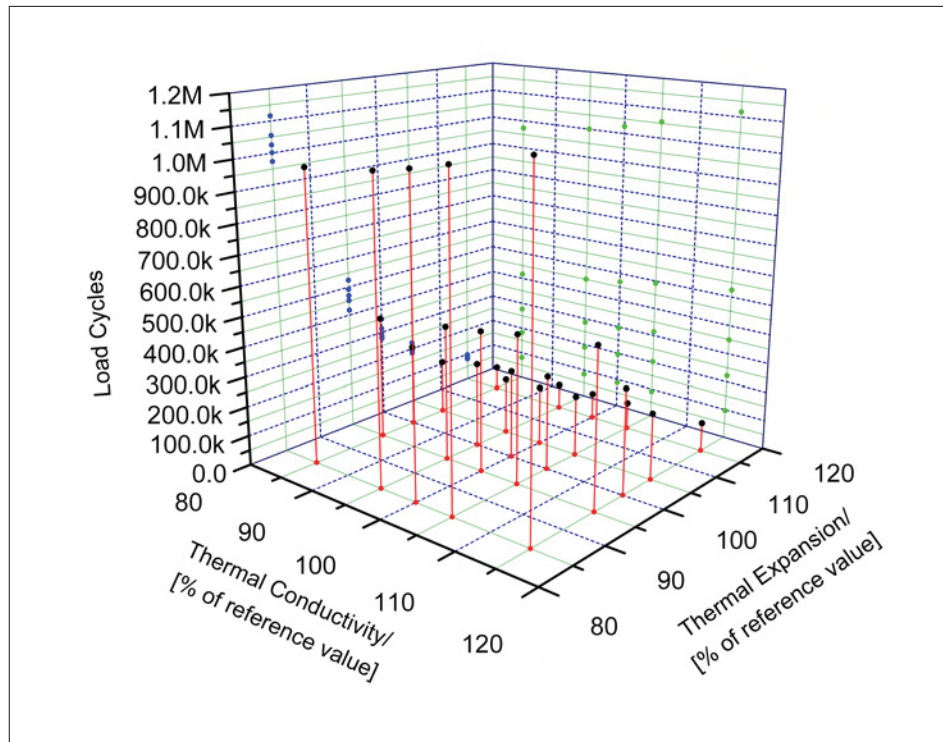


**Figure 6.8:** X38CrMoV5-1 47 HRC: Load cycles until a crack length of 1.5 mm is reached in dependence of  $\alpha$  for fixed values of  $\lambda$ .

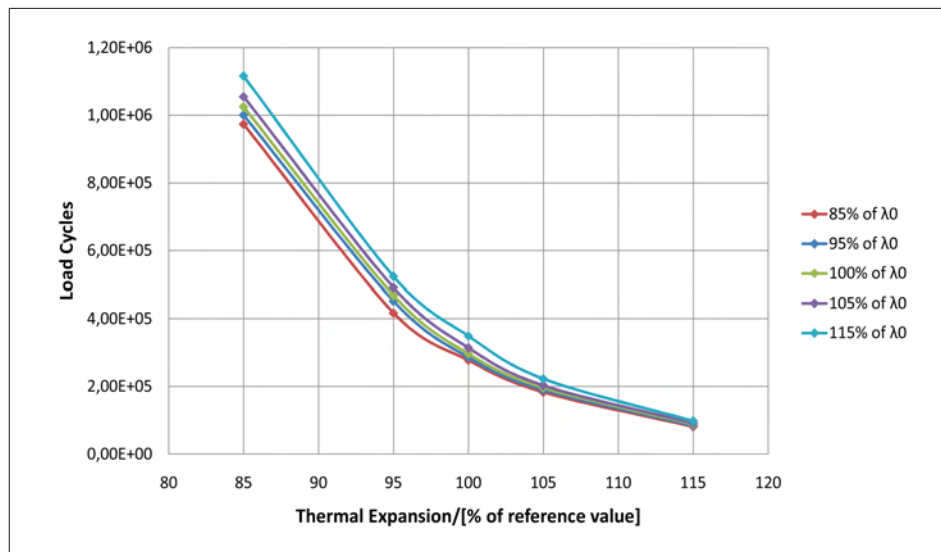


**Figure 6.9:** X38CrMoV5-1 47 HRC: Load cycles until a crack length of 1.5 mm is reached in dependence of  $\lambda$  for fixed values of  $\alpha$ .

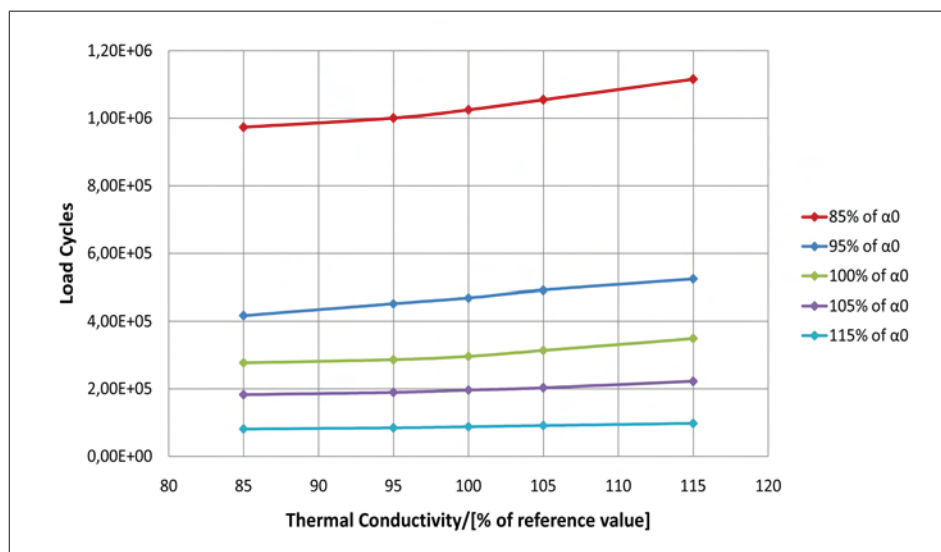




**Figure 6.10:** X38CrMoV5-1 50 HRC: Load cycles until a crack length of 1.5 mm is reached.

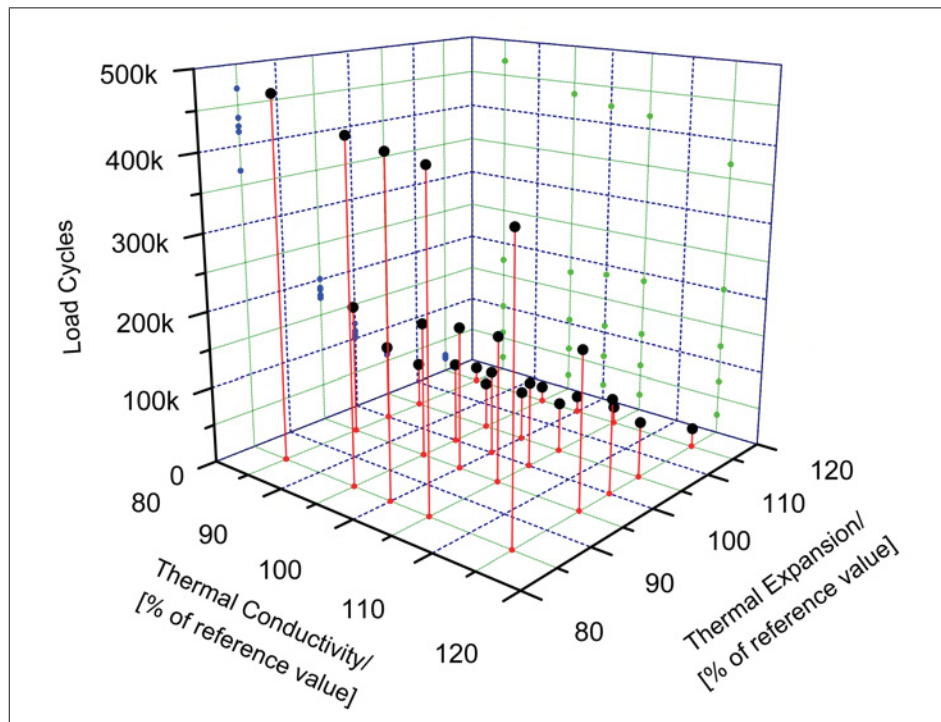


**Figure 6.11:** X38CrMoV5-1 50 HRC: Load cycles until a crack length of 1.5 mm is reached in dependence of  $\alpha$  for fixed values of  $\lambda$ .

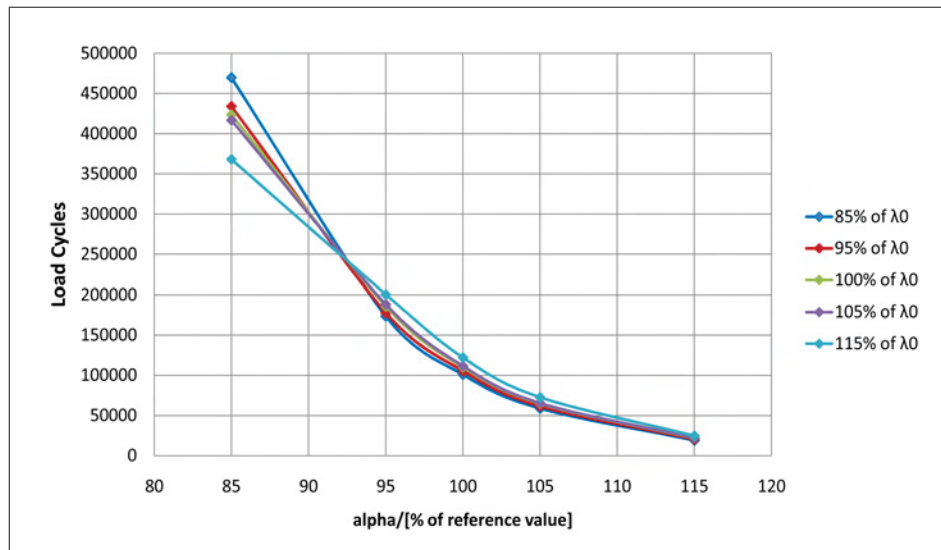


**Figure 6.12:** X38CrMoV5-1 50 HRC: Load cycles until a crack length of 1.5 mm is reached in dependence of  $\lambda$  for fixed values of  $\alpha$ .

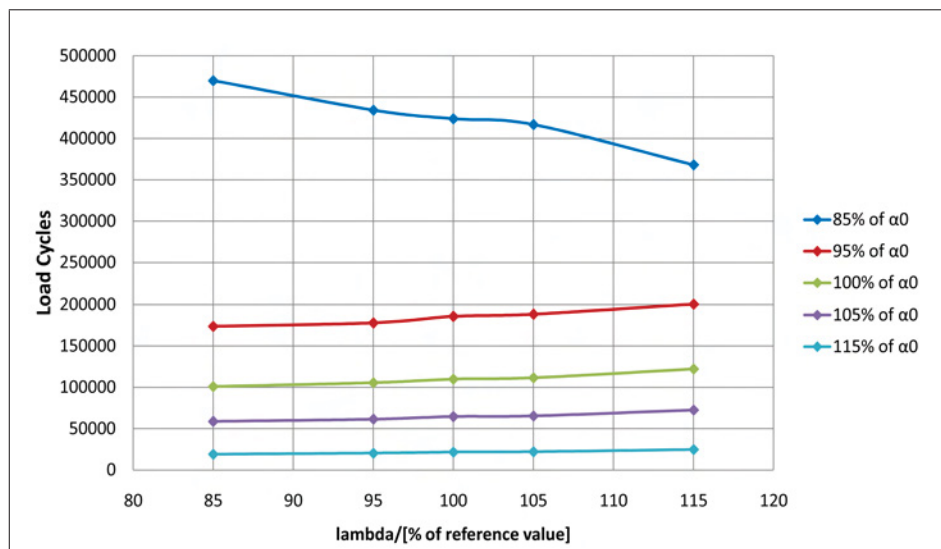
### Influence of Material Parameters on the Evolution of Crack Arrays with Variable Spacing-Detailed Results for 44 HRC



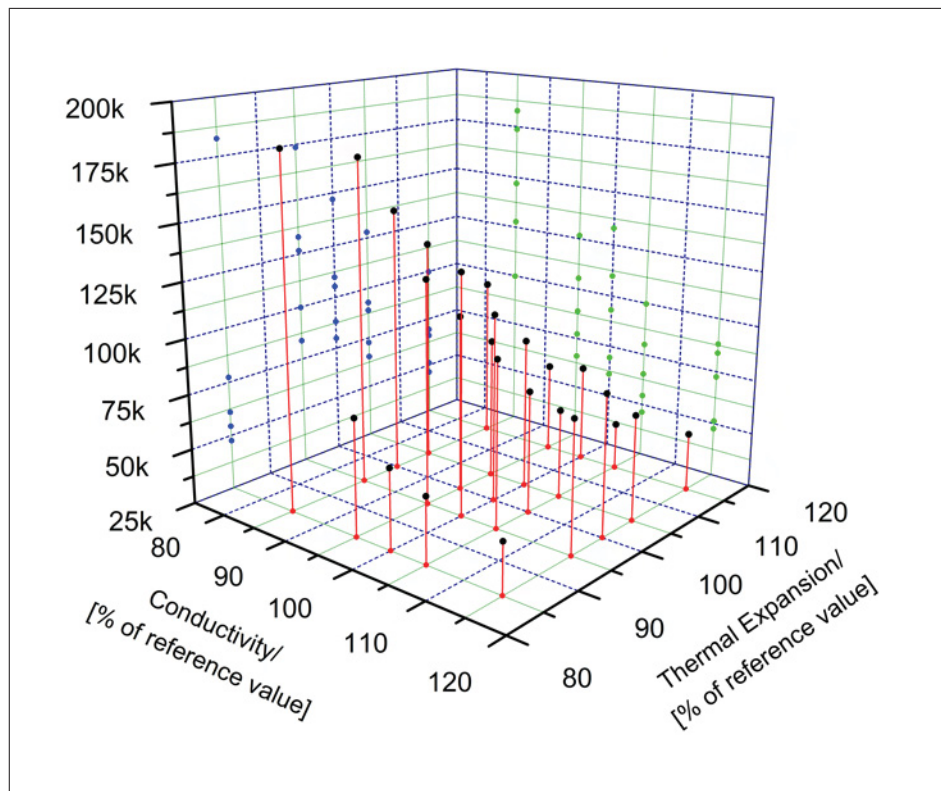
**Figure 6.13:** Load cycles until a crack length of 1.5 mm is reached including variable crack spacing based on the observed standard spacing 2.25 mm and a 10% influence of the driving force. Data on a X38CrMoV5-1 hot work tool steel with a hardness of 44 HRC.



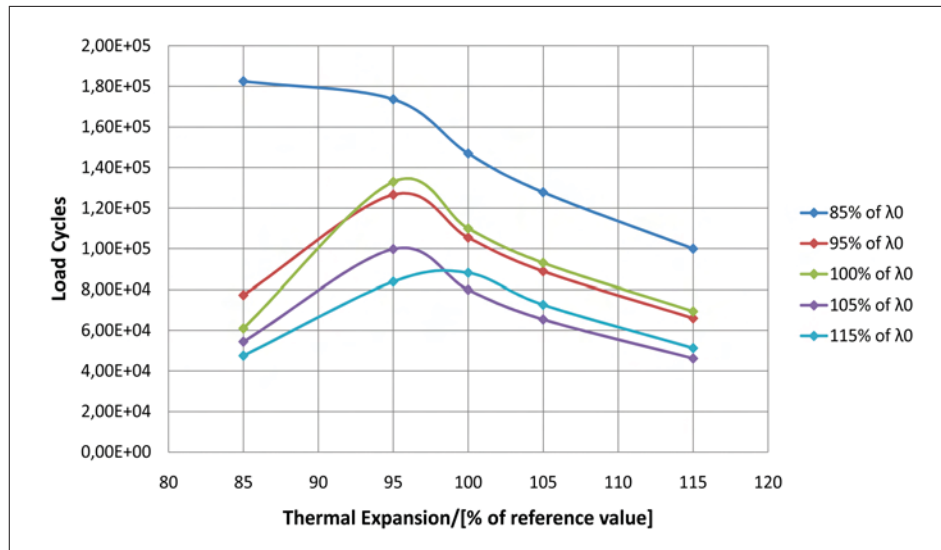
**Figure 6.14:** Load cycles until a crack length of 1.5 mm is reached based on the observed standard spacing 2.25 mm and a 10% influence of the driving force in dependence of  $\alpha$  for fixed values of  $\lambda$ .



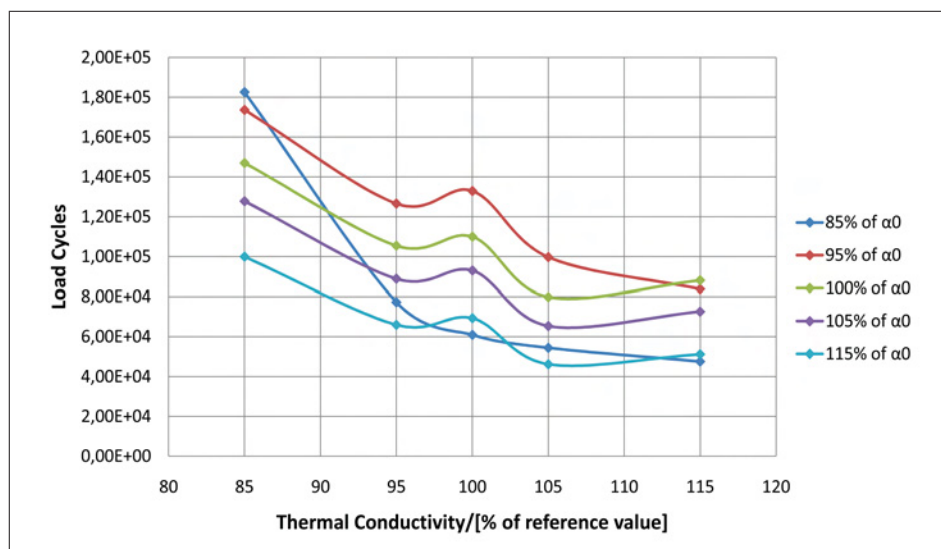
**Figure 6.15:** Load cycles until a crack length of 1.5 mm is reached based on the observed standard spacing 2.25 mm and a 10% influence of the driving force in dependence of  $\lambda$  for fixed values of  $\alpha$ .



**Figure 6.16:** Load cycles until a crack length of 1.5 mm is reached including variable crack spacing based on the observed standard spacing 2.25 mm and a 100% influence of the driving force. Data on a X38CrMoV5-1 hot work tool steel with a hardness of 44 HRC.



**Figure 6.17:** Load cycles until a crack length of 1.5 mm is reached based on the observed standard spacing 2.25 mm and a 100% influence of the driving force in dependence of  $\alpha$  for fixed values of  $\lambda$ .



**Figure 6.18:** Load cycles until a crack length of 1.5 mm is reached based on the observed standard spacing 2.25 mm and a 100% influence of the driving force in dependence of  $\lambda$  for fixed values of  $\alpha$ .

## Control Script for generation of C-Models

```

1 #####
2 #Settings and Loops
3 #####
4
5 #Import of user defined modules
6 import MyData
7 import MyFunctions
8 import MySettings
9
10 #Import of Python modules
11 from Numeric import *
12 import os
13 import time
14 t1=time.clock()
15
16 #Import of Abaqus Python modules
17 from abaqus import *
18 from abaqusConstants import *
19 from caeModules import *
20 from part import *
21 from material import *
22 from section import *
23 from assembly import *
24 from step import *
25 from interaction import *
26 from load import *
27 from mesh import *
28 from job import *
29 from sketch import *
30 from visualization import *
31 from connectorBehavior import *
32
33 #Basic settings:
34 Mode=1
35 CS_IteratorEnd=0
36 CP_IteratorStart=0
37 CP_IteratorEnd=100
38 Study_IteratorStart=1
39 Study_IteratorEnd=1
40 SIF_CorrectionStart=0
41 SIF_CorrectionEnd=1
42 CrackPropagationsAlt=0
43 Threshold=3.0
44 C_Slope=1.14786E-9
45 m_Exponent=5.2091
46 JobName_Root='CrackPropagation'
47 mdb_close='On'
48 Stabilize='Off'
49 BorderExpansion=False
50 NLGEOM=OFF
51 CPUs=2
52
53 #For input
54 Variations_H=["44HRC", "47HRC", "50HRC"]
55 Variations_L=["+0","+5", "+15", "-5", "-15"]
56 Variations_A=["+0","+5", "+15", "-5", "-15"]
57
58 #For C-Model
59 Variations_A_Coeff=[1.00, 1.05, 1.15, 0.95, 0.85]
60 #For output
61 Variations_L_Percent=["100", "105", "115", "95", "85"]
62 Variations_A_Percent=["100", "105", "115", "95", "85"]
63
64 MODEL='C-Model'
65 PART='EvenDieArea'
66 firstRun_finished=False
67
68 #Declaration of lists which must be available in the entire study.
69 Array_SIFs=[]
70 Array_SIFs_closed=[]
71 Array_initialSpacing=[]
72 CrackLengths=[]
73 CrackSpacings=[]
74 DataSets_Labels=[]
75 DataSets_Counter=-1
76 DataSets_Total=len(Variations_H)*len(Variations_L)*len(Variations_A)
77
78 for Iterator_H in range(len(Variations_H)):
79     for Iterator_L in range(len(Variations_L)):
80         for Iterator_A in range(len(Variations_A)):
81             DataSets_Counter+=1
82             CrackLengths.append([])
83             CrackSpacings.append([])
84
85             for IteratorIndex_Study in range(Study_IteratorStart, Study_IteratorEnd+1):
86                 Break=False

```

```

88 BreakCounter=0
89
90 for Iterator_Spacing in range(CS_IteratorEnd+1):
91
92     #Initial configuration of crack array
93     Elements=1
94     NumberOfCracks=1
95     Propagation_Increment=0.05
96     UnitCell=[0]
97     CrackDistance1, UnitCell[0]=MyData.InitialData(Variations_H[Iterator_H], Variations_L[Iterator_L], ←
          Variations_A[Iterator_A])
98     CrackLength1=UnitCell[0]
99     UnitCell_copy=copy.deepcopy(UnitCell)
100    CrackLengths[DataSets_Counter].append([])
101    DeltaKeff_Array=[]
102    LoadCycles_Array=[]
103    Cycles_total=0
104    InstabilityNumber=0
105    InstabilityBoolean=False
106
107    if Break:
108        break
109    else:
110        for Iterator_CrackProp in range(CP_IteratorStart, CP_IteratorEnd+1):
111
112            if Break:
113                break
114            else:
115                for Iterator_CP_alternative in range(CrackPropagationsAlt+1):
116                    for SIF_Correction in range(SIF_CorrectionStart, SIF_CorrectionEnd+1):
117
118                        if SIF_Correction==0:
119                            Mode=1
120                        if SIF_Correction==1:
121                            Mode='1_closed'
122
123                        SequenceNr0=Variations_H[Iterator_H]+'_L'+Variations_L[Iterator_L]+'_A'+Variations_A[Iterator_A]
124                        SequenceNr1=str(IteratorIndex_Study)
125                        SequenceNr2=str(SIF_Correction)
126                        SequenceNr3=str(Iterator_Spacing)
127                        SequenceNr4=str(Iterator_CrackProp)
128                        SequenceNr5=str(Iterator_CP_alternative)
129                        #In order to get clearly arranged data in the end, the labels of all data sets need to be stored←
130
131                        if Iterator_Spacing==0 and Iterator_CrackProp==0:
132                            DataSets_Labels.append(SequenceNr0)
133                            JobNameDataSet=JobName_Root+'-DataSet_'+SequenceNr0
134                            Jobname=JobName_Root+'-DataSet_'+SequenceNr0+'_ID-G_'+SequenceNr1+'_SIF_'+SequenceNr2+'←
          _SequenceNumber'+SequenceNr3+'-'+SequenceNr4+'-'+SequenceNr5
135                            # "PathDat" ist nur fuer Risswachstum relevant. Da die Daten der vorhergehenden Rechnung ←
          eingelesen werden lautet die 3. SequenceNr: Iterator_CP_alternative-1
136                            PathDat=Jobname+'.dat'
137
138                            #Path to the User Subroutines
139                            NameSubroutine='/sim/Peter.R.57/Subroutines/Subroutines_fertig/Step-999_fertig/←
          Subroutine_Utemp_Sigini2_Step-999_'+Variations_H[Iterator_H]+'_L'+Variations_L[Iterator_L]+←
          '_A'+Variations_A[Iterator_A]+'_f'
140
141
142                        #Declaration of lists which must be available in a single loop.
143                        All_CP_Directions=[]
144                        AllCrackSeams=[]
145                        K_Factors_Sets=[]
146
147                        #####
148                        #Dimensions and Seed Properties
149                        #####
150                        Radius, SeedSize_fine, SeedSize_fine_top, SeedSize_coarse, SeedSize_Flanks, Transition_Bias, ←
          Transition_Elements, SegmentsCircle, Circles, BiasCircles, TransitionDistance_rel, ←
          TransitionWidth_rel = \
151                        MySettings.StandardSettings(NumberOfCracks, CrackLength1)
152                        UnitCell, UnitCell_copy, Cracks_defined, CrackSpacing, BorderSpacing, NumberOfCracks, ←
          CrackLengths, CrackSpacings, ModelWidth, Radius, SeedSize_Flanks, SeedSize_fine_top, ←
          SeedSize_fine, SeedSize_coarse, TransitionDistance_abs, TransitionWidth_abs, ←
          Transition_Bias, Transition_Elements, ModelHeight, Elements = \
153                        MySettings.ModeSettings(Mode, Iterator_CrackProp, Iterator_Spacing, CrackDistance1, ←
          NumberOfCracks, CrackLengths, DataSets_Counter, UnitCell, UnitCell_copy, CrackSpacings, ←
          Elements, InstabilityBoolean, Propagation_Increment)
154
155                        #####
156                        #Begin Part Level
157                        #####
158                        mdb.models.changeKey(fromName='Model-1', toName=MODEL)
159                        mdb.models[MODEL].ConstrainedSketch(name='__profile__', sheetSize=200.0)
160                        a0=mdb.models[MODEL]
161                        a=mdb.models[MODEL].sketches['__profile__']
162                        #SC... shortcut
163                        a.LineSC=a.Line
164
165                        if NumberOfCracks==0:
166                            PointsSketch=[(0.0, 0.0), (ModelWidth_noCracks, 0.0), (ModelWidth_noCracks, -ModelHeight), (←

```



```

165         ModelWidth_noCracks/2, -ModelHeight), (0.0, -ModelHeight), (0.0, 0.0)]
166     for i in range(len(PointsSketch)-1):
167         a_LineSC(point1=PointsSketch[i], point2=PointsSketch[i+1])
168
169     elif NumberOfCracks>0:
170         PointsSketch=[(0.0, 0.0), (BorderSpacing, 0.0)]
171         for i in range(NumberOfCracks-1):
172             PointsSketch.append((BorderSpacing+CrackSpacing*i, 0.0))
173             PointsSketch.append((BorderSpacing+CrackSpacing*(i+1), 0.0))
174         PointsSketch+=[(BorderSpacing+CrackSpacing*(NumberOfCracks-1), 0.0), (BorderSpacing+
175             CrackSpacing*(NumberOfCracks-1)+BorderSpacing, 0.0),
176             (BorderSpacing+CrackSpacing*(NumberOfCracks-1)+BorderSpacing, -ModelHeight), ((BorderSpacing+
177             CrackSpacing*(NumberOfCracks-1)+BorderSpacing)/2, -ModelHeight), (0.0, -ModelHeight),
178             (0.0, 0.0)]
179
180         for i in range(len(PointsSketch)-1):
181             a_LineSC(point1=PointsSketch[i], point2=PointsSketch[i+1])
182
183     #Part is created
184     a0.Part(dimensionality=TWO_D_PLANAR, name=PART, type=DEFORMABLE_BODY)
185     a0.parts[PART].BaseShell(sketch=a)
186     del a
187
188     #Partitioning
189     a0.ConstrainedSketch(gridSpacing=5.0, name='__profile__', sheetSize=200.0, transform=a0.parts[
190     PART].MakeSketchTransform(
191     sketchPlane=a0.parts[PART].faces[0], sketchPlaneSide=SIDE1, sketchOrientation=RIGHT, origin=
192     (0.0, 0.0, 0.0))
193     a2=a0.sketches['__profile__']
194     b1=a0.parts[PART]
195     b1.projectReferencesOntoSketch(filter=COPLANAR_EDGES, sketch=a0.sketches['__profile__'])
196     #SC...shortcut
197     a2_LineSC=a2.Line
198     a2_verticesSC=a2.vertices
199
200     NumberOfUnitCells=int(NumberOfCracks/Elements)
201     Rest=NumberOfCracks%Elements
202     if NumberOfCracks>0:
203
204         #Circles are inserted.
205         #Periodicities will be considered.
206         if NumberOfUnitCells==0:
207             Iterator1=1
208             Iterator2=NumberOfCracks
209         else:
210             Iterator1=NumberOfUnitCells
211             Iterator2=Elements
212         for i in range(Iterator1):
213             for j in range(Iterator2):
214                 a2_LineSC(point1=(BorderSpacing+CrackSpacing*j+CrackSpacing*(Elements)*i, 0),
215                 point2=(BorderSpacing+CrackSpacing*j+CrackSpacing*(Elements)*i, -UnitCell[j]))
216                 if Rest!=0 and i==NumberOfUnitCells-1 and j==Elements-1:
217                     ZusatzRisse=NumberOfCracks-NumberOfUnitCells*Elements
218                     k=0
219                     #This loop gets active in case an incomplete unit cell needs to be generated.
220                     while k<ZusatzRisse:
221                         j=j+1
222                         a2_LineSC(point1=(BorderSpacing+CrackSpacing*j+CrackSpacing*(Elements)*i, 0),
223                         point2=(BorderSpacing+CrackSpacing*j+CrackSpacing*(Elements)*i, -L))
224                         k=k+1
225
226         #Alternative crack propagation (MTS criterion)
227         AllCrackTipCoord=[]
228         QVector_StartPoints=[]
229         #Within this loop all data must be available. Thus some lists are declared outside the loop.
230         #QVector_StartPoints: Together with AllCrackTipCoord QVectors for crack definition can be
231         defined.
232         if Iterator_CP_alternative>0:
233             if Rest==0:
234                 Iterator_End1=NumberOfUnitCells
235             else:
236                 Iterator_End1=NumberOfUnitCells+1
237             for i in range(Iterator_End1):
238                 if i<NumberOfUnitCells:
239                     Iterator_End2=Elements
240                 else:
241                     Iterator_End2=NumberOfCracks-Elements*NumberOfUnitCells
242             for j in range(Iterator_End2):
243                 X_Vertex1=[]
244                 Y_Vertex1=[]
245                 X_Vertex2=[]
246                 Y_Vertex2=[]
247                 X_Vertex1.append(BorderSpacing+CrackSpacing*j+CrackSpacing*Elements*i)
248                 Y_Vertex1.append(-UnitCell[j])
249                 X_Vertex2.append(BorderSpacing+CrackSpacing*j+CrackSpacing*Elements*i+
250                 CrackLength_ParisLaw*sin(radians(All_CP_Directions[0][j+Elements*i])))
251                 Y_Vertex2.append(-UnitCell[j]-CrackLength_ParisLaw*cos(radians(All_CP_Directions[0][j+
252                 Elements*i])))
253             #Initial cracks and crackpropagations are partitioned for each crack.
254             for k in range(1, Iterator_CP_alternative+1):
255                 if k==1:

```

```

246         a2_LineSC(point1=(X_Vertex1[k-1], Y_Vertex1[k-1]), point2=(X_Vertex2[k-1], Y_Vertex2[←
[k-1]))
247     Angle=0
248     for m in range(k):
249         Angle=Angle+All_CP_Directions[m][j+Elements*i]
250     if k>1:
251         X_Vertex1.append(X_Vertex2[k-2])
252         Y_Vertex1.append(Y_Vertex2[k-2])
253         #The newly determined angles refer to the actual propagation.
254         # => In order to draw the new propagation all angles need to be added!
255         X_Vertex2.append(X_Vertex2[k-2]+CrackLength_ParisLaw*sin(radians(Angle)))
256         Y_Vertex2.append(Y_Vertex2[k-2]-CrackLength_ParisLaw*cos(radians(Angle)))
257         a2_LineSC(point1=(X_Vertex1[k-1], Y_Vertex1[k-1]), point2=(X_Vertex2[k-1],
258         Y_Vertex2[k-1]))
259     #Circles around the crack tips
260     if k==Iterator_CP_alternative:
261         Radius=((X_Vertex2[Iterator_CP_alternative-1]-X_Vertex1[Iterator_CP_alternative-1])←
**2+(Y_Vertex2[Iterator_CP_alternative-1]-Y_Vertex1[Iterator_CP_alternative-1])←
**2)**0.5
262     a2.CircleByCenterPerimeter(center=(X_Vertex2[Iterator_CP_alternative-1], Y_Vertex2[←
Iterator_CP_alternative-1]), point1=(X_Vertex2[Iterator_CP_alternative-1]-←
Radius*sin(radians(Angle)), Y_Vertex2[Iterator_CP_alternative-1]+Radius*cos(←
radians(Angle)))
263     #SIN and COS always have the right sign.
264     #Crack tip coordinates are written to a list for set definitions later on.
265     #When the loop passes these lines the list gets defined anew and the old values are ←
gone.
266     CrackTipCoord=[]
267     CrackTipCoord.append(X_Vertex2[Iterator_CP_alternative-1])
268     CrackTipCoord.append(Y_Vertex2[Iterator_CP_alternative-1])
269     AllCrackTipCoord.append(CrackTipCoord)
270     QVector_a=[]
271     QVector_a.append(X_Vertex2[Iterator_CP_alternative-1]-Radius*sin(radians(Angle)))
272     QVector_a.append(Y_Vertex2[Iterator_CP_alternative-1]+Radius*cos(radians(Angle)))
273     QVector_StartPoints.append(QVector_a)
274     Seam=[]
275     Seam.append(X_Vertex2[Iterator_CP_alternative-1]-(CrackLength_ParisLaw/2)*sin(←
radians(Angle)))
276     Seam.append(Y_Vertex2[Iterator_CP_alternative-1]+(CrackLength_ParisLaw/2)*cos(←
radians(Angle)))
277     AllCrackSeams.append(Seam)
278     #Partitioning of the circles if there are initiating cracks only.
279     if Iterator_CP_alternative==0:
280         if Rest==0:
281             Iterator_End1=NumberOfUnitCells
282         else:
283             Iterator_End1=NumberOfUnitCells+1
284         for i in range(Iterator_End1):
285             if i<NumberOfUnitCells:
286                 Iterator_End2=Elements
287             else:
288                 Iterator_End2=NumberOfCracks-Elements*NumberOfUnitCells
289             for j in range(Iterator_End2):
290                 X_Vertex1=(BorderSpacing+CrackSpacing*j+CrackSpacing*Elements*i)
291                 Y_Vertex1=(-UnitCell[j])
292                 a2.CircleByCenterPerimeter(center=(X_Vertex1, Y_Vertex1), point1=(X_Vertex1, Y_Vertex1+←
Radius))
293                 CrackTipCoord=[]
294                 CrackTipCoord.append(X_Vertex1)
295                 CrackTipCoord.append(Y_Vertex1)
296                 AllCrackTipCoord.append(CrackTipCoord)
297                 QVector_a=[]
298                 #In case of a straight crack X_Vertex1 and X_Vertex2 are equal.
299                 QVector_a.append(X_Vertex1)
300                 QVector_a.append(Y_Vertex1+UnitCell[j])
301                 QVector_StartPoints.append(QVector_a)
302                 Seam=[]
303                 Seam.append(X_Vertex1)
304                 #The order of the vertices is important. The first vertex is part of the edge above the ←
circle.
305                 #The second vertex is situated at the tip, i.e. it is part of the edge within the ←
circle.
306                 Seam.append(-0.001)
307                 Seam.append(Y_Vertex1)
308                 AllCrackSeams.append(Seam)
309     #Partitioning of transition area
310     a2_LineSC(point1=(0.0, -TransitionDistance_rel), point2=(ModelWidth, -TransitionDistance_rel))
311     a2_LineSC(point1=(0.0, -TransitionWidth_rel), point2=(ModelWidth, -TransitionWidth_rel))
312     b1.PartitionFaceBySketch(faces=b1.faces, sketch=a2)
313     #Sketch for partitioning is finished
314     del a2
315     #####
316     #Properties
317     #####
318     #b1.faces gives a tuple with the labels of all faces.
319     b1.Set(name='Set_AllFaces', faces=b1.faces)
320     a0.Material(name='W300')
321     a0.materials['W300'].Elastic(table=(
322     (210000.0, 0.3, 20.0),
323     (199000.0, 0.3, 100.0),

```

```

324         (194500.0, 0.3, 200.0),
325         (190000.0, 0.3, 300.0),
326         (188000.0, 0.3, 400.0),
327         (183000.0, 0.3, 500.0),
328         (183000.0, 0.3, 600.0),
329         (183000.0, 0.3, 700.0)),
330         temperatureDependency=ON)
331     a0.materials['W300'].Expansion(table=(
332         (10.5*10**-6*Variations_A_Coeff[Iterator_A], 20.0),
333         (11.1*10**-6*Variations_A_Coeff[Iterator_A], 95.0),
334         (11.67*10**-6*Variations_A_Coeff[Iterator_A], 185.0),
335         (12.25*10**-6*Variations_A_Coeff[Iterator_A], 291.0),
336         (12.72*10**-6*Variations_A_Coeff[Iterator_A], 397.0),
337         (13.14*10**-6*Variations_A_Coeff[Iterator_A], 502.0),
338         (13.36*10**-6*Variations_A_Coeff[Iterator_A], 613.0)),
339         temperatureDependency=ON)
340     a0.HomogeneousSolidSection(material='W300', name='Section_EvenDieArea', thickness=1.0)
341     b1.SectionAssignment(offset=0.0, region=b1.sets['Set_AllFaces'], sectionName='↵
        Section_EvenDieArea')
342 #####
343 #Begin Assembly Level
344 #####
345     a0.rootAssembly.Instance(dependent=OFF, name='EvenDieArea-1', part=b1)
346     c1=a0.rootAssembly
347     c2=c1.instances['EvenDieArea-1']
348     c1_SetSC=c1.Set
349     c1_setsSC=c1.sets
350     c2_verticesSC=c2.vertices
351     c2_edgesSC=c2.edges
352     c2_facesSC=c2.faces
353     c2_verticesSC_findAtSC=c2_verticesSC.findAt
354     c2_edgesSC_findAtSC=c2_edgesSC.findAt
355     c2_facesSC_findAtSC=c2_facesSC.findAt
356 #####
357 #Cracks and Seams
358 #####
359 #Each crack tip is stored to a set.
360 for i in range(NumberOfCracks):
361     c1_SetSC(name='CTip'+str(i+1), vertices=c2_verticesSC_findAtSC(((AllCrackTipCoord[i][0], ↵
        AllCrackTipCoord[i][1], 0), ), ))
362 #
363 #Crack definition
364 #
365 for i in range(NumberOfCracks):
366     if Cracks_defined[i]==True:
367         c1_engineeringFeatures.ContourIntegral(collapsedElementAtTip=SINGLE_NODE, crackFront=↵
            c1_setsSC['CTip'+str(i+1)], crackTip=c1_setsSC['CTip'+str(i+1)], ↵
            extensionDirectionMethod=
368             Q_VECTORS, midNodePosition=0.25, name='Crack-'+str(i+1), qVectors=((QVector_StartPoints[i↵
                ][0], QVector_StartPoints[i][1], 0.0), (AllCrackTipCoord[i][0], AllCrackTipCoord[i][1],↵
                0.0)), ), symmetric=OFF)
369 #
370 #Assign Seam
371 #
372 #The crack flanks are stored in sets and defined as seams.
373 a0.ContactProperty('IntProp-1')
374 a0.interactionProperties['IntProp-1'].TangentialBehavior(formulation=FRICTIONLESS)
375 #The default setting Penalty does not yield satisfying results. Lagrange is better in this case.
376 a0.interactionProperties['IntProp-1'].NormalBehavior(allowSeparation=ON, ↵
            clearanceAtZeroContactPressure=0.0,
377             constraintEnforcementMethod=AUGMENTED_LAGRANGE, contactStiffness=DEFAULT, ↵
            contactStiffnessScaleFactor=1.0, pressureOverclosure=HARD)
378 if Iterator_CP_alternative==0:
379     for i in range(NumberOfCracks):
380         if Cracks_defined[i]==True:
381             #j runs from 1 to 2 because the circle splits the crack in two.
382             for j in range(1,3):
383                 c1_SetSC(name='CrackFlank'+str(i+1)+str(j), edges=c2_edgesSC_findAtSC(((AllCrackSeams[i↵
                    ][0], AllCrackSeams[i][j], 0), ), ))
384                 c1_engineeringFeatures.assignSeam(regions=c1_setsSC['CrackFlank'+str(i+1)+str(j)])
385 #Contact Definitions
386 #Small Sliding is activated
387 for i in range(NumberOfCracks):
388     if Cracks_defined[i]==True:
389         c1.Surface(name='Surface'+str(i+1), side12Edges=(c1_setsSC['CrackFlank'+str(i+1)+str(1)].↵
            edges, c1_setsSC['CrackFlank'+str(i+1)+str(2)].edges))
390         a0.SelfContactStd(createStepName='Initial', enforcement=SURFACE_TO_SURFACE, ↵
            interactionProperty='IntProp-1', name='Kontakt'+str(i+1), surface=c1_surfaces['↵
            Surface'+str(i+1)], thickness=ON)
391 else:
392     for i in range(NumberOfCracks+ NumberOfCracks*Iterator_CP_alternative):
393         c1_SetSC(name='CrackFlank'+str(i+1), edges=c2_edgesSC_findAtSC(((AllCrackSeams[i][0], ↵
            AllCrackSeams[i][1], 0), ), ))
394         c1_engineeringFeatures.assignSeam(regions=c1_setsSC['CrackFlank'+str(i+1)])
395 #Contact Definitions
396 c1.Surface(name='Surface'+str(i+1), side12Edges=(c1_setsSC['CrackFlank'+str(i+1)].edges))
397 a0.SelfContactStd(createStepName='Initial', enforcement=SURFACE_TO_SURFACE, ↵
            interactionProperty='IntProp-1', name='Kontakt'+str(i+1)+str(j), surface=c1_surfaces['↵
            Surface'+str(i+1)], thickness=ON)

```

```

399 #####
400 #Step
401 #####
402 a0.StaticStep(initialInc=0.0001, maxInc=1.0, maxNumInc=1000000, minInc=1e-08, description='', ←
name='MainStep', nlgeom=NLGEOM, convertSDI=CONVERT_SDI_ON, previous='Initial')
403 #####
404 #Boundary Conditions: Fixed and Loose Bearing
405 #####
406 a0.DisplacementBC_SC=a0.DisplacementBC
407 if NumberOfCracks==0:
408     c1_SetSC(name='Set_FixedBearing', vertices=c2_verticesSC_findAtSC(((ModelWidth_noCracks/2, ←
ModelHeight, 0), ), ))
409     c1_SetSC(name='Set_Bottom_left', edges=c2_edgesSC_findAtSC(((ModelWidth_noCracks/2-0.001, ←
ModelHeight, 0), ), ))
410     c1_SetSC(name='Set_Bottom_right', edges=c2_edgesSC_findAtSC(((ModelWidth_noCracks/2+0.001, ←
ModelHeight, 0), ), ))
411 if NumberOfCracks>0:
412     c1_SetSC(name='Set_FixedBearing', vertices=c2_verticesSC_findAtSC(((BorderSpacing+←
CrackSpacing*(NumberOfCracks-1)+BorderSpacing)/2, -ModelHeight, 0), ), ))
413     c1_SetSC(name='Set_Bottom_left', edges=c2_edgesSC_findAtSC(((BorderSpacing/2, -ModelHeight, 0)←
), ))
414     c1_SetSC(name='Set_Bottom_right', edges=c2_edgesSC_findAtSC(((1.5*BorderSpacing+CrackSpacing←
*(NumberOfCracks-1)), -ModelHeight, 0), ), ))
415 a0.DisplacementBC_SC(amplitude=UNSET, createStepName='Initial', distributionType=UNIFORM, ←
fieldName='', fixed=OFF, localCsys=None, name='FixedBearing', region=c1_setsSC['←
Set_FixedBearing'], u1=0.0, u2=0.0, ur3=UNSET)
416 a0.DisplacementBC_SC(amplitude=UNSET, createStepName='Initial', distributionType=UNIFORM, ←
fieldName='', fixed=OFF, localCsys=None, name='Bottom_left', region=c1_setsSC['←
Set_Bottom_left'], u1=UNSET, u2=0.0, ur3=UNSET)
417 a0.DisplacementBC_SC(amplitude=UNSET, createStepName='Initial', distributionType=UNIFORM, ←
fieldName='', fixed=OFF, localCsys=None, name='Bottom_right', region=c1_setsSC['←
Set_Bottom_right'], u1=UNSET, u2=0.0, ur3=UNSET)
418 #####
419 #History/Field Output Request
420 #####
421 for i in range(NumberOfCracks):
422     a0.HistoryOutputRequest(contourIntegral='Crack-'+str(i+1), contourType=K_FACTORS, ←
createStepName='MainStep', frequency=LAST_INCREMENT, name='K_Factor'+str(i+1), ←
numberOfContours=5, rebar=EXCLUDE, sectionPoints=DEFAULT)
423     a0.fieldOutputRequests['F-Output-1'].setValues(variables=('S', 'MISESMAX', 'U', 'RF', 'NT'))
424 #####
425 #Seeds + Mesh
426 #####
427 c1_seedEdgeBySizeSC=c1.seedEdgeBySize
428 c1_seedEdgeByBiasSC=c1.seedEdgeByBias
429 c1_seedEdgeByNumberSC=c1.seedEdgeByNumber
430 if NumberOfCracks==0:
431     #fine meshed area
432     c1_SetSC(name='fine_top', edges=c2_edgesSC_findAtSC(((ModelWidth_noCracks/2, 0), ), ))
433     c1_SetSC(name='fine_top3', edges=c2_edgesSC_findAtSC(((ModelWidth_noCracks/2, ←
TransitionDistance_rel, 0), ), ))
434     c1_seedEdgeBySizeSC(constraint=FINER, edges=c1_setsSC['fine_top'].edges, size=SeedSize_fine)
435     #Fine mesh and a fixed number of elements for periodic boundary conditions
436     c1_SetSC(name='fine_fixed_left', edges=c2_edgesSC_findAtSC(((0, -0.001, 0), ), ))
437     c1_SetSC(name='fine_fixed_right', edges=c2_edgesSC_findAtSC(((ModelWidth_noCracks, -0.001, 0), ←
), ))
438     c1_seedEdgeBySizeSC(constraint=FIXED, edges=c1_setsSC['fine_fixed_left'].edges, size=←
SeedSize_fine)
439     c1_seedEdgeBySizeSC(constraint=FIXED, edges=c1_setsSC['fine_fixed_right'].edges, size=←
SeedSize_fine)
440     #FINER
441     c1_SetSC(name='coarse_top', edges=c2_edgesSC_findAtSC(((1.0, -TransitionWidth_rel, 0), ), ))
442     c1_SetSC(name='coarse_bottom_left', edges=c2_edgesSC_findAtSC(((ModelWidth_noCracks/2-0.001, ←
ModelHeight, 0), ), ))
443     c1_SetSC(name='coarse_bottom_right', edges=c2_edgesSC_findAtSC(((ModelWidth_noCracks ←
/2+0.001, -ModelHeight, 0), ), ))
444     #FIXED
445     c1_SetSC(name='coarse_fixed_left', edges=c2_edgesSC_findAtSC(((0, -ModelHeight+0.001, 0), ), ))
446     c1_SetSC(name='coarse_fixed_right', edges=c2_edgesSC_findAtSC(((ModelWidth_noCracks, ←
ModelHeight+0.001, 0), ), ))
447     c1_seedEdgeBySizeSC(constraint=FINER, edges=c1_setsSC['coarse_top'].edges, size=←
SeedSize_coarse)
448     c1_seedEdgeBySizeSC(constraint=FINER, edges=c1_setsSC['coarse_bottom_left'].edges, size=←
SeedSize_coarse)
449     c1_seedEdgeBySizeSC(constraint=FINER, edges=c1_setsSC['coarse_bottom_right'].edges, size=←
SeedSize_coarse)
450     c1_seedEdgeBySizeSC(constraint=FINER, edges=c1_setsSC['coarse_bottom_right'].edges, size=←
SeedSize_coarse)
451     c1_seedEdgeBySizeSC(constraint=FIXED, edges=c1_setsSC['coarse_fixed_left'].edges, size=←
SeedSize_coarse)
452     c1_seedEdgeBySizeSC(constraint=FIXED, edges=c1_setsSC['coarse_fixed_right'].edges, size=←
SeedSize_coarse)
453     #Transition area with bias
454     #Mind the difference between end2Edges and end1Edges!
455     c1_seedEdgeByBiasSC(constraint=FIXED, end2Edges=c2_edgesSC_findAtSC(((0, ←
TransitionDistance_rel-0.001, 0), ), ), number=Transition_Elements, ratio=Transition_Bias)
456     c1_seedEdgeByBiasSC(constraint=FIXED, end1Edges=c2_edgesSC_findAtSC(((ModelWidth_noCracks, ←
TransitionDistance_rel-0.001, 0), ), ), number=Transition_Elements, ratio=Transition_Bias)
457 if NumberOfCracks>0:
458     #fine meshed area
459     #Line above transition area

```

```

460     c1_SetSC(name= 'fine_top1', edges=c2_edgesSC_findAtSC(((BorderSpacing/2,-←
461         TransitionDistance_rel, 0), ),))
462     #Lines on top of the plate
463     c1_SetSC(name= 'fine_top2', edges=c2_edgesSC_findAtSC(((BorderSpacing/2,0, 0), ),))
464     c1_SetSC(name= 'fine_top3', edges=c2_edgesSC_findAtSC(((1.5*BorderSpacing+CrackSpacing*(←
465         NumberOfCracks-1),0, 0), ),))
466     for i in range(NumberOfCracks-1):
467         c1_SetSC(name= 'fine_top'+str(i+4), edges=c2_edgesSC_findAtSC(((BorderSpacing/2+CrackSpacing←
468             *(i+1),0, 0), ),))
469     #Mesh size at the top of the plate. Since the mesh must fit the one in the crack areas it ←
470     #should be finer than SeedSize_fine.
471     for i in range(1, NumberOfCracks+2):
472         c1_seedEdgeBySizeSC(constraint=FINER, edges=c1_setsSC['fine_top'+str(i+1)].edges, size=←
473             SeedSize_fine_top)
474         c1_seedEdgeBySizeSC(constraint=FINER, edges=c1_setsSC['fine_top1'].edges, size=SeedSize_fine)
475     #Fine mesh and a fixed number of elements for periodic boundary conditions
476     c1_SetSC(name= 'fein_fixed_left', edges=c2_edgesSC_findAtSC(((0,-0.001, 0), ),))
477     c1_SetSC(name= 'fein_fixed_right', edges=c2_edgesSC_findAtSC(((2*BorderSpacing+CrackSpacing*(←
478         NumberOfCracks-1),-0.001, 0), ),))
479     c1_seedEdgeBySizeSC(constraint=FIXED, edges=c1_setsSC['fein_fixed_left'].edges, size=←
480         SeedSize_fine)
481     c1_seedEdgeBySizeSC(constraint=FIXED, edges=c1_setsSC['fein_fixed_right'].edges, size=←
482         SeedSize_fine)
483     #FINER
484     c1_SetSC(name= 'coarse_top', edges=c2_edgesSC_findAtSC(((0.001,-TransitionWidth_rel, 0), ),))
485     c1_SetSC(name= 'coarse_bottom_left', edges=c2_edgesSC_findAtSC(((BorderSpacing/2,-ModelHeight,←
486         0), ),))
487     c1_SetSC(name= 'coarse_bottom_right', edges=c2_edgesSC_findAtSC(((1.5*BorderSpacing+←
488         CrackSpacing*(NumberOfCracks-1),-ModelHeight, 0), ),))
489     #FIXED
490     c1_SetSC(name= 'coarse_fixed_left', edges=c2_edgesSC_findAtSC(((0,-ModelHeight+0.001, 0), ),))
491     c1_SetSC(name= 'coarse_fixed_right', edges=c2_edgesSC_findAtSC(((2*BorderSpacing+CrackSpacing←
492         *(NumberOfCracks-1),-ModelHeight+0.001, 0), ),))
493     c1_seedEdgeBySizeSC(constraint=FINER, edges=c1_setsSC['coarse_top'].edges, size=←
494         SeedSize_coarse)
495     c1_seedEdgeBySizeSC(constraint=FINER, edges=c1_setsSC['coarse_bottom_left'].edges, size=←
496         SeedSize_coarse)
497     c1_seedEdgeBySizeSC(constraint=FINER, edges=c1_setsSC['coarse_bottom_right'].edges, size=←
498         SeedSize_coarse)
499     c1_seedEdgeBySizeSC(constraint=FIXED, edges=c1_setsSC['coarse_fixed_left'].edges, size=←
500         SeedSize_coarse)
501     c1_seedEdgeBySizeSC(constraint=FIXED, edges=c1_setsSC['coarse_fixed_right'].edges, size=←
502         SeedSize_coarse)
503     #Transition area with bias
504     #Mind the difference between end2Edges and end1Edges!
505     c1_seedEdgeByBiasSC(constraint=FIXED, end2Edges=c2_edgesSC_findAtSC(((0,-←
506         TransitionDistance_rel-0.001, 0), ),), number=Transition_Elements, ratio=Transition_Bias)
507     c1_seedEdgeByBiasSC(constraint=FIXED, end1Edges=c2_edgesSC_findAtSC(((2*BorderSpacing+←
508         CrackSpacing*(NumberOfCracks-1),-TransitionDistance_rel-0.001, 0), ),), number=←
509         Transition_Elements, ratio=Transition_Bias)
510     #Circles
511     for i in range(NumberOfCracks):
512         if Iterator_CP_alternative==0:
513             c1_seedEdgeByNumberSC(constraint=FINER, edges=c2_edgesSC_findAtSC(((AllCrackTipCoord[i][0]+←
514                 Radius, AllCrackTipCoord[i][1],0), ),), number=SegmentsCircle)
515             c1_seedEdgeByBiasSC(end2Edges=c2_edgesSC_findAtSC(((AllCrackTipCoord[i][0],AllCrackTipCoord[←
516                 i][1],0), ),), number=Circles, ratio=BiasCircles)
517         else:
518             c1_seedEdgeByNumberSC(constraint=FINER, edges=c2_edgesSC_findAtSC(((QVector_StartPoints[i]←
519                 [0]+2*(AllCrackTipCoord[i][0]-QVector_StartPoints[i][0]), QVector_StartPoints[i]←
520                 [1]+2*(AllCrackTipCoord[i][1]-QVector_StartPoints[i][1]),0), ),), number=←
521                 SegmentsCircle)
522             c1_seedEdgeByBiasSC(end2Edges=c2_edgesSC_findAtSC(((AllCrackTipCoord[i][0],AllCrackTipCoord[←
523                 i][1],0), ),), number=Circles, ratio=BiasCircles)
524     #Finer Constraint crack flanks outside the circular region.
525     for i in range(1,NumberOfCracks+1):
526         if Cracks_defined[i-1]==True:
527             c1_seedEdgeBySizeSC(constraint=FINER, edges=c1_setsSC['CrackFlank'+str(i)+'1'].edges, size=←
528                 SeedSize_Flanks)
529     #The Mesh Controls of the whole model are set to QUAD. Afterwards the coarse area and transition ←
530     #area are set to TRI and the sections around the crack tips are set to Quad-Dominated.
531     c1_SetSC(name='CoarseArea_Set', faces=c2_facesSC_findAtSC(((0, -ModelHeight,0), ),))
532     c1_SetSC(name='TransitionArea_Set', faces=c2_facesSC_findAtSC(((0, -TransitionDistance_rel ←
533         -0.0001,0), ),))
534     c1.setMeshControls(elemShape=QUAD, regions=c2.faces)
535     c1.setMeshControls(elemShape=TRI, regions=c1_setsSC['CoarseArea_Set'].faces)
536     c1.setMeshControls(elemShape=TRI, regions=c1_setsSC['TransitionArea_Set'].faces)
537     for i in range(NumberOfCracks):
538         c1.setMeshControls(elemShape=QUAD_DOMINATED, regions=c2.faces.findAt(((AllCrackTipCoord[i][0],←
539             AllCrackTipCoord[i][1], 0), ),), technique=SWEEP)
540     c1_SetSC(name= 'AllFaces', faces=c2.faces)
541     c1.setElementType(elemTypes=(ElemType(elemCode=CPE8, elemLibrary=STANDARD), ElemType(elemCode=←
542         CPE6M, elemLibrary=STANDARD)), regions=c1_setsSC['AllFaces'])
543     c1.generateMesh(regions=(c2, ))
544     #####
545     #Job
546     #####
547     mdb.Job(contactPrint=OFF, description='', echoPrint=OFF, explicitPrecision=SINGLE, historyPrint=←
548         OFF, memory=50, memoryUnits=PERCENTAGE, model=MODEL, modelPrint=OFF, multiprocessingMode=←

```

```

                    DEFAULT, name=Jobname, nodalOutputPrecision=SINGLE, numCpus=CPUs, numDomains=CPUs, ←
                    parallelizationMethodExplicit=DOMAIN, scratch='', type=ANALYSIS, userSubroutine=←
                    NameSubroutine)
519 #####
520 #Periodic Boundary Conditions
521 #####
522 #Opposite nodes must be linked by equation-constraints.
523 #Since the order of the labels is arbitrary, the corresponding pairs need to be identified by ←
    their y-coordinates.
524 #No pair can be linked more than once per degree of freedom. Thus, duplicate pairs originating ←
    from the vertices between edges must be removed.
525 #In this script the nodes are are stored to a text file with inp-file syntax
526 if NumberOfCracks==0:
527     c1_SetSC(edges=c2_edgesSC_findAtSC(((0,-0.001,0),),), name='Edge_left1')
528     c1_SetSC(edges=c2_edgesSC_findAtSC(((ModelWidth_noCracks,-0.001,0),),), name='Edge_right1')
529     c1_SetSC(edges=c2_edgesSC_findAtSC(((0,-1*(TransitionDistance_rel+TransitionWidth_abs/2),0),) ←
    ),), name='Edge_left2')
530     c1_SetSC(edges=c2_edgesSC_findAtSC(((ModelWidth_noCracks,-1*(TransitionDistance_rel+ ←
    TransitionWidth_abs/2),0),),), name='Edge_right2')
531     c1_SetSC(edges=c2_edgesSC_findAtSC(((0,-1*(ModelHeight-0.001),0),),), name='Edge_left3')
532     c1_SetSC(edges=c2_edgesSC_findAtSC(((ModelWidth_noCracks,-1*(ModelHeight-0.1),0),),), name=' ←
    Edge_right3')
533 if NumberOfCracks>0:
534     c1_SetSC(edges=c2_edgesSC_findAtSC(((0,-0.001,0),),), name='Edge_left1')
535     c1_SetSC(edges=c2_edgesSC_findAtSC(((NumberOfCracks-1)*CrackSpacing+2*BorderSpacing,-0.001,0) ←
    ),), name='Edge_right1')
536     c1_SetSC(edges=c2_edgesSC_findAtSC(((0,-1*(TransitionDistance_rel+TransitionWidth_abs/2),0),) ←
    ),), name='Edge_left2')
537     c1_SetSC(edges=c2_edgesSC_findAtSC(((NumberOfCracks-1)*CrackSpacing+2*BorderSpacing,-1*( ←
    TransitionDistance_rel+TransitionWidth_abs/2),0),),), name='Edge_right2')
538     c1_SetSC(edges=c2_edgesSC_findAtSC(((0,-1*(ModelHeight-0.001),0),),), name='Edge_left3')
539     c1_SetSC(edges=c2_edgesSC_findAtSC(((NumberOfCracks-1)*CrackSpacing+2*BorderSpacing,-1*( ←
    ModelHeight-0.001),0),),), name='Edge_right3')
540     c1_SetSC(nodes=c1_setsSC['Edge_left1'].nodes, name='AL')
541     c1_SetSC(nodes=c1_setsSC['Edge_right1'].nodes, name='AR')
542     c1_SetSC(nodes=c1_setsSC['Edge_left2'].nodes, name='BL')
543     c1_SetSC(nodes=c1_setsSC['Edge_right2'].nodes, name='BR')
544     c1_SetSC(nodes=c1_setsSC['Edge_left3'].nodes, name='CL')
545     c1_SetSC(nodes=c1_setsSC['Edge_right3'].nodes, name='CR')
546
547 #Node labels and their y-coordinates are stored to arrays.
548 LabelCoord1L=[(n.label, n.coordinates[1]) for n in c1_setsSC['AL'].nodes]
549 LabelCoord1R=[(n.label, n.coordinates[1]) for n in c1_setsSC['AR'].nodes]
550 LabelCoord2L=[(n.label, n.coordinates[1]) for n in c1_setsSC['BL'].nodes]
551 LabelCoord2R=[(n.label, n.coordinates[1]) for n in c1_setsSC['BR'].nodes]
552 LabelCoord3L=[(n.label, n.coordinates[1]) for n in c1_setsSC['CL'].nodes]
553 LabelCoord3R=[(n.label, n.coordinates[1]) for n in c1_setsSC['CR'].nodes]
554
555 LabelCoordL=LabelCoord1L+LabelCoord2L+LabelCoord3L
556 LabelCoordR=LabelCoord1R+LabelCoord2R+LabelCoord3R
557
558 #The two arrays are sorted by y-coordinates.
559 LabelCoordL.sort(lambda x, y: cmp(x[1],y[1]))
560 LabelCoordR.sort(lambda x, y: cmp(x[1],y[1]))
561
562 NodePairs=len(LabelCoordL)
563 CounterKW=0
564 file = open("AddKeywordsNSETS_"+JobName_Root+".inp", "w")
565 file.write('\nNSET,nset=LeftX, instance=EvenDieArea-1, Unsorted\n')
566 for i in range(NodePairs):
567     if LabelCoordL[i][1]==LabelCoordL[i-1][1]:
568         continue
569     else:
570         CounterKW+=1
571         if CounterKW==10:
572             file.write('\n')
573             CounterKW=0
574         file.write(str(LabelCoordL[i][0]))
575         if i < NodePairs-1:
576             file.write(',')
577 file.write('\nNSET,nset=LeftY, instance=EvenDieArea-1, Unsorted\n')
578 CounterKW=0
579 for i in range(NodePairs):
580     if LabelCoordL[i][1]==LabelCoordL[i-1][1] or LabelCoordL[i][1]==(-ModelWidth):
581         continue
582     else:
583         CounterKW+=1
584         if CounterKW==10:
585             file.write('\n')
586             CounterKW=0
587         file.write(str(LabelCoordL[i][0]))
588         if i < NodePairs-1:
589             file.write(',')
590 file.write('\nNSET,nset=RightX, instance=EvenDieArea-1, Unsorted\n')
591 CounterKW=0
592 for i in range(NodePairs):
593     if LabelCoordR[i][1]==LabelCoordR[i-1][1]:
594         continue
595     else:
596         CounterKW+=1

```

```

597         if CounterKW==10:
598             file.write('\n')
599             CounterKW=0
600         file.write(str(LabelCoordR[i][0]))
601         if i < NodePairs-1:
602             file.write(',')
603     file.write('\n*NSET,nset=RightY, instance=EvenDieArea-1, Unsorted\n')
604     CounterKW=0
605     for i in range(NodePairs):
606         if LabelCoordR[i][1]==LabelCoordR[i-1][1] or LabelCoordR[i][1]==(-ModelWidth):
607             continue
608         else:
609             CounterKW+=1
610             if CounterKW==10:
611                 file.write('\n')
612                 CounterKW=0
613             file.write(str(LabelCoordR[i][0]))
614             if i < NodePairs-1:
615                 file.write(',')
616     file.close()
617
618     if BorderExpansion==False:
619         MyFunctions.DatenAusgabe('*Equation\n2\nLeftX, 1, 1., RightX, 1, -1.\n*Equation\n2\nLeftY, 2, ←
        1., RightY, 2, -1.', "AddKeywordsEQUATIONS_"+JobName_Root+".inp", "w")
620     if BorderExpansion==True:
621         #Reference Points for consideration of thermal expansion
622         Counter=0
623         refPoints=[]
624         for i in range(NodePairs):
625             if i==0:
626                 r1 = c1.referencePoints
627             if LabelCoordL[i][1]==LabelCoordL[i-1][1]:
628                 continue
629             else:
630                 c1.ReferencePoint(point=(-1.0, LabelCoordL[i][1], 0.0))
631                 #The reference points are added at the first position of the repository.
632                 refPoints.append(r1.values()[0])
633                 #If the reference point is fixed it has no influence in the equations.
634                 c1_SetSC(referencePoints=(refPoints[Counter],), name='RF-'+str(Counter))
635                 yPosition=LabelCoordL[i][1]
636
637                 #Heat expansion is calculated with a temperature curve fit from Mathematica and the ←
638                 thermal expansion coefficient at 95°C.
639                 Temperature=MyData.T_Distributions(Variations_L[Iterator_L], Iterator_L, yPosition)
640                 BorderExpansion_Value=(11.1*10**-6)*(Temperature-120)*ModelWidth
641                 a0.DisplacementBC(amplitude=UNSET, createStepName='MainStep', distributionType=UNIFORM, ←
642                 fieldName='', fixed=OFF, localCsys=None, name='BC-'+str(Counter), region=Region(←
643                 referencePoints=(refPoints[Counter],), u1=BorderExpansion_Value, u2=UNSET, ur3=UNSET←
644                 )
645                 Counter+=1
646
647         #For the periodic boundary conditions in x-direction the nodes need to be defined in single ←
648         sets.
649         file = open("AddKeywordsNSETS.inp", "a")
650         Counter=0
651         for i in range(NodePairs):
652             if LabelCoordL[i][1]==LabelCoordL[i-1][1]:
653                 continue
654             else:
655                 file.write('\n*NSET,nset=LeftX-'+str(Counter)+'', instance=EvenDieArea-1, Unsorted\n')
656                 file.write(str(LabelCoordL[i][0]))
657                 Counter+=1
658         Counter=0
659         for i in range(NodePairs):
660             if LabelCoordR[i][1]==LabelCoordR[i-1][1]:
661                 continue
662             else:
663                 file.write('\n*NSET,nset=rightX-'+str(Counter)+'', instance=EvenDieArea-1, Unsorted\n')
664                 file.write(str(LabelCoordR[i][0]))
665                 Counter+=1
666     file.close()
667
668     #The coefficient of the Reference Points must be chosen as 1. This results from the definition ←
669     of the boundary conditions of the RFs.
670     MyFunctions.DatenAusgabe('', "AddKeywordsEQUATIONS.inp", "w")
671     for i in range(Counter):
672         MyFunctions.DatenAusgabe('*Equation\n3\nLeftX-'+str(i)+'', 1, 1., RightX-'+str(i)+'', 1, -1., ←
673         RF-'+str(i)+'', 1, 1.\n', "AddKeywordsEQUATIONS.inp", "a")
674     MyFunctions.DatenAusgabe('*Equation\n2\nLeftY, 2, 1., RightY, 2, -1.', "AddKeywordsEQUATIONS.←
675     inp", "a")
676     #####
677     #Initial Conditions
678     #####
679     # a0.Temperature causes the implementation of the Subroutine UTEMP.
680     # Alternatively, UTEMP could be called by adding "+TEMPERATURE, USER\nallNodes" in the keyword ←
681     editor (the set 'allNodes' needs to be created first).
682     c1_SetSC(name='CompleteEvenDieArea', faces=c2.faces)
683     a0.Temperature(createStepName='MainStep', distributionType=USER_DEFINED, name='TemperatureField'←
684     , region=c1_setsSC['CompleteEvenDieArea'])
685

```

```

676 #The initial temperature of the prewarmed die is 120 centigrade.
677 a0.Temperature(CreateStepName='Initial', crossSectionDistribution=CONSTANT_THROUGH_THICKNESS, ←
distributionType=UNIFORM, magnitudes=(120.0, ), name='InitialTemperature', region=c1_setsSC←
['CompleteEvenDieArea'])
678 #####
679 #Update of Keywords
680 #####
681 #"*INCLUDE" is used for adding the data on the periodic BC to the inp-file.
682 #Call of Sigini is not possible in CAE. It has to be added to the inp-file via "*INITIAL ←
CONDITIONS" in the KW editor.
683 #"*Energy Print" is an inp-file command causing the energy data to be written to the dat-file.
684 #FREQUENCY=999999 causes the energy to be written to file after the last increment only.
685 #Within the keywords a characteristic expression is sought which is used to insert the ←
additional keywords in the right place.
686 # "synchVersions" must be positioned in the code above the lines changing the keywords.
687 a0.keywordBlock.synchVersions(storeNodesAndElements=False)
688 for mem in a0.keywordBlock.sieBlocks:
689     if mem.startswith('*End Instance'):
690         memID_NSET = a0.keywordBlock.sieBlocks.index(mem)
691     elif mem.startswith('*Step,'):
692         memID_Initial = a0.keywordBlock.sieBlocks.index(mem)
693     elif mem.startswith('*End Step'):
694         memID_Energy = a0.keywordBlock.sieBlocks.index(mem)
695 a0.keywordBlock.insert(memID_NSET, '\n*INCLUDE, Input=AddKeywordsNSETS'+JobName_Root+'.inp ')
696 a0.keywordBlock.insert(memID_NSET+1, '\n*INCLUDE, Input=AddKeywordsEQUATIONS'+JobName_Root+'.←
inp')
697 a0.keywordBlock.insert(memID_Initial, '\n*INITIAL CONDITIONS, TYPE=STRESS, USER')
698 a0.keywordBlock.insert(memID_Energy+2, '\n*ENERGY PRINT, FREQUENCY=9999999')
699 if Stabilize=='0n':
700     a0.keywordBlock.insert(memID_Energy+3, '\n*CONTACT CONTROLS, STABILIZE')
701 #####
702 #Job Definition and Submission
703 #####
704 #Either the job can be started or just the inp-file can be created.
705 #mdb.jobs[Jobname].writeInput(consistencyChecking=OFF)
706 mdb.jobs[Jobname].submit(consistencyChecking=OFF)
707
708 #The loop will proceed after analysis has finished.
709 mdb.jobs[Jobname].waitForCompletion()
710 #The current mdb will be closed. This way all keywords will be deleted and potential errors will←
be avoided.
711 if mdb_close=='0n':
712     mdb.close()
713     firstRun_finished=True
714 #####
715 #Begin Output Processing and Generation
716 #####
717 K_Factors, SIF_Crack1, CP_Directions=MyFunctions.K_Output(PathDat, NumberOfCracks)
718
719 if Iterator_CP_alternative>0 or Mode==1 or Mode=='1_simpelCrackPropagation':
720
721     #Data are systematically written to an array.
722     if Iterator_Spacing==0 and Iterator_CrackProp==0:
723         Array_SIFs.append([])
724         Array_initialSpacing.append([])
725     if Iterator_CrackProp==0:
726         Array_SIFs[DataSets_Counter].append([])
727         Array_initialSpacing[DataSets_Counter].append([])
728     if Iterator_CrackProp>0:
729         Array_SIFs[DataSets_Counter][Iterator_Spacing].append(K_Factors)
730
731     K_Factors_Sets.append(K_Factors)
732     All_CP_Directions.append(CP_Directions)
733
734     #The output of studies on crack spacing is treated differently.
735     if IteratorIndex_Study==0:
736         if Array_SIFs[DataSets_Counter][Iterator_Spacing][Iterator_CrackProp][0] >= Threshold:
737
738             FileToBeRead = open("Module_CrackDistance.txt", "a")
739             FileToBeRead.write('\n#'+Jobname)
740             FileToBeRead.write('\nif HRC_F=="'+str(Variations_H[Iterator_H])+'" and Lambda_F=="'+str(←
Variations_L[Iterator_L])+'" and Alpha_F=="'+str(Variations_A[Iterator_A])+'"')
741             FileToBeRead.write('\t\tCrackDistance="'+str(CrackDistance)+'\n')
742             FileToBeRead.write('\t\tInitialCrackLength="'+str(UnitCell[0])+''\n')
743             dFileToBeRead.close()
744
745             if os.path.exists("CrackDistance.txt")==False:
746                 FileToBeRead2 = open("CrackDistance.txt", "a")
747                 FileToBeRead2.write('\nHRC\tLambda\tAlpha\tDistance\tLength\n')
748                 FileToBeRead2.close()
749                 FileToBeRead2 = open("CrackDistance.txt", "a")
750                 FileToBeRead2.write('\n'+str(Variations_H[Iterator_H])+'\t'+str(Variations_L[Iterator_L])+←
'\t'+str(Variations_A[Iterator_A])+'\t'+str(RissAbstand)+'\t'+str(UnitCell[0]))
751                 FileToBeRead2.close()
752
753             Array_initialSpacing[DataSets_Counter][Iterator_Spacing].append([Array_SIFs[←
DataSets_Counter][Iterator_Spacing][VIterator_CrackProp][0]])
754             if (DataSets_Counter+1)==DataSets_Total:
755                 #All SIFs are written to a text file
756                 MyModule.Output_Distance(DataSets_Total, CS_IteratorEnd, CP_IteratorEnd, Array_SIFs, ←

```



```

DataSets_Labels, CrackLength1, Propagation_Increment, CrackSpacings, '↵
KFactors_Function'+Jobname+'.txt')
757 #Threshold data is written to a file.
758 MyModule.Output_Distance(DataSets_Total, CS_IteratorEnd, CP_IteratorEnd, ↵
Array_initialSpacing, DataSets_Labels, CrackLength1, Propagation_Increment, ↵
CrackSpacings, 'InitialSpacing'+Jobname+'.txt')

759 Break=True
760 break
761 else:
762     Array_initialSpacing[DataSets_Counter][Iterator_Spacing].append([0])
763
764 if Mode=='1_closed':
765     if Iterator_Spacing==0 and Iterator_CrackProp==0:
766         Array_SIFs_closed.append([])
767     if Iterator_CrackProp==0:
768         Array_SIFs_closed[DataSets_Counter].append([])
769     if Iterator_CrackProp>0:
770         Array_SIFs_closed[DataSets_Counter][Iterator_Spacing].append(K_Factors)
771 #####
772 #Calculation of DeltaKeff
773 #####
774 #Sigma_max => access to current odb
775 Sigma_max=MyFunctions.ReadOdb(Jobname, NODAL, 'S', 'S11', 'CTIP1')
776 #Sigma_min => access to MyData
777 Sigma_min=MyData.S_Envelopes(Variations_H[Iterator_H], Variations_L[Iterator_L], Variations_A[↵
Iterator_A], (-UnitCell[0]))
778 #Stress Ratio
779 StressRatio=Sigma_min/Sigma_max
780 #Correction Factor
781 CorrectionFactor=MyData.DeltaKeff_Calculation(StressRatio)
782 #####
783 #Criterion for Instability
784 #####
785 DeltaKeff=Array_SIFs[DataSets_Counter][Iterator_Spacing][Iterator_CrackProp][0]*↵
CorrectionFactor
786 DeltaKeff_Array.append(DeltaKeff)
787
788 if DeltaKeff < Threshold:
789     InstabilityNumber+=1
790     InstabilityBoolean=True
791     BreakCounter+=1
792     if DeltaKeff >= Threshold:
793         BreakCounter=0
794 #####
795 #Integration of Crack Propagation Curve
796 #####
797 PropagationRate=C_Slope*(DeltaKeff)**m_Exponent
798 if Iterator_CrackProp==0:
799     Circles=0
800     LoadCycles_Array.append(0)
801 else:
802     Circles=MyFunctions.PropagationFunction(DeltaKeff_Array[-1],DeltaKeff_Array[-2], ↵
CrackLengths[0][0][-1], CrackLengths[0][0][-2], C_Slope, m_Exponent)
803     LoadCycles_Array.append(Circles)
804     Cycles_total+=Circles
805 #####
806 #Main Output Files
807 #####
808 if Iterator_CrackProp==0:
809     MyFunctions.DatenAusgabe('Crack Length\tSigma_min\tSigma_max\tR\tFactor\tSIF Crack1\↵
tDeltaKeff\tInstabilities\tPropagationRate\tCycles\tCycles_total\n', 'A_StressRatios'+↵
JobNameDataSet+'-CD-'+str(CrackDistance1)+'.txt', 'a')
810 MyFunctions.DatenAusgabe(JobNameDataSet+'\nCrack Length\tLoadCycles\n', 'A_LoadCycles.txt', ↵
'a')
811 MyFunctions.DatenAusgabe('%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s' % (str(UnitCell[0]), '↵
\t', str(Sigma_min), '\t', str(Sigma_max), '\t', str(StressRatio), '\t', str(↵
CorrectionFactor), '\t', str(SIF_Crack1), '\t', str(DeltaKeff), '\t', str(BreakCounter), '\t↵
', str(PropagationRate), '\t', str(Circles), '\t', str(Cycles_total), '\n'), '↵
A_StressRatios'+JobNameDataSet+'-CD-'+str(CrackDistance1)+'.txt', 'a')
812 MyFunctions.DatenAusgabe('%s%s%s' % (str(UnitCell[0]), '\t', str(Cycles_total), '\n'), '↵
A_LoadCycles.txt', 'a')
813
814 if BreakCounter==3 or UnitCell[0]>1.5 or Iterator_CrackProp==CP_IteratorEnd:
815     if (DataSets_Counter+1)==DataSets_Total:
816         MyFunctions.Output_Growth(DataSets_Total, CS_IteratorEnd, CP_IteratorEnd, Array_SIFs, ↵
DataSets_Labels, CrackLengths, CrackSpacings, 'A_SIFs_CrackPropagation_'+↵
JobNameDataSet+'-CD-'+str(CrackDistance1)+'.txt')
817         MyFunctions.Output_Growth(DataSets_Total, CS_IteratorEnd, CP_IteratorEnd, ↵
Array_SIFs_closed, DataSets_Labels, CrackLengths, CrackSpacings, 'A_SIFs_closedModel_↵
'+JobNameDataSet+'-CD-'+str(CrackDistance1)+'.txt')
818     Break=True
819     break
820     t2=time.clock()
821     MyFunctions.DatenAusgabe('\n'+str(t2-t1), "Time.txt", "a")
822     #In order to save disk space the current files are deleted.
823     os.remove(Jobname+'.com')
824     os.remove(Jobname+'.inp')
825     os.remove(Jobname+'.sta')
826     os.remove(Jobname+'.msg')
827     os.remove(Jobname+'.prt')
828     os.remove(Jobname+'.odb')

```