Simulation of Complex Microstructural Geometries using X-FEM and the Application to Solder Joint Lifetime Prediction

Alexander Menk
(Dipl.-Ing.)

Submitted in fulfillment of the requirements
for the
Degree of PhD

Department of Civil Engineering
School of Engineering
University of Glasgow

April 18, 2011

Abstract

In electronic devices solder joints form a mechanical as well as an electrical connection between the circuit board and the component (e.g. a chip or a resistor). Temperature variations occurring during field use cause crack initiation and crack growth inside the joints. Accurate prediction of the lifetime requires a method to simulate the damage process based on microstructural properties.

Numerical simulation of developing cracks and microstructural entities such as grain boundaries and grain junctions gives rise to several problems. The solution contains strong and weak discontinuities as well as weak singularities. To obtain reasonable solutions with the finite element method (FEM) the element edges have to align with the cracks and the grain boundaries, which imposes geometrical restrictions on the mesh choice. Additionally, a large number of elements has to be used in the vicinity of the singularities which increases the computational effort. Both problems can be circumvented with the extended finite element method (X-FEM) by using appropriate enrichment functions.

In this thesis the X-FEM will be developed for the simulation of complex microstructural geometries. Due to the anisotropy of the different grains forming a joint and the variety of different microstructural configurations it is not always possible to write the enrichment functions in a closed form. A procedure to determine enrichment functions numerically is explained and tested. As a result, a very simple meshing scheme, which will be introduced here, can be used to simulate developing cracks in solder joint microstructures. Due to the simplicity of the meshing algorithm the simulation can

be automated completely. A large number of enrichment functions must be used to realize this. Well-conditioned equation systems, however, cannot be guaranteed for such an approach. To improve the condition number of the X-FEM stiffness matrix and thus the robustness of the solution process a preconditioning technique is derived and applied.

This approach makes it possible to develop a new and fully automated procedure for addressing the reliability of solder joints numerically. The procedure relies on the random generation of microstructures. Performing crack growth calculations for a series of these structures makes it possible to address the influence of varying microstructures on the damage process. Material parameters describing the microstructure are determined in an inverse procedure. It will be shown that the numerical results correspond well with experimental observations.

Contents

List of tables

List of figures

Acknowledgements

Writing this thesis would not have been possible without the support and the help of the people around me. First of all I would like to thank Professor Stéphane P. A. Bordas for the supervision of this thesis, for the knowledge he supplied and the inspiring discussions we had. Furthermore, I would like to thank Professor Chris Pearce for supervising the final part of this thesis. I would also like to express my gratitude to the Robert Bosch GmbH for funding this work and for making this kind of research possible. Especially I would like to thank Olivier Lanier, Roumen Ratchev, Markus Klingler, Michael Guyenot, Erik Deier and Andreas Fix at the research department of Bosch for the fruitful discussions.
Last but most importantly I would like to thank my girlfriend Kathrin Kokernak who constantly supported me throughout the process and I would like to dedicate this thesis to her.

Declaration

I declare that this thesis is a record of the original work carried out by myself under the supervision of Prof. S. P. A. Bordas and Professor Chris Pearce in the Department of Civil Engineering at the University of Glasgow, United Kingdom, and at the research department of the Robert Bosch GmbH, Germany, during the period of April 2008 to March 2011. The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright acts. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis. The thesis has not been presented elsewhere in consideration for a higher degree.

Alexander Menk

Notations

$a(\cdot, \cdot)$	Bilinear form
$a_1, ..., a_n$	Entries of the coefficient vector
a_{ik}	Coefficients of the shape functions
a_{ijk}^X	Coefficients of the enrichment functions
$\boldsymbol{a}$	Coefficient vector
$\boldsymbol{a}_i$	Iterates of an iterative method
$\mathcal{A}$	Operator on a Hilbert space
$\boldsymbol{b}$	Body forces
$\boldsymbol{B}$	Matrix used to describe the numerical enrichment procedure
$D_{\Gamma_C}(\boldsymbol{x})$	Distance function
$\boldsymbol{D}_{ijkl}$	Fourth order elasticity tensor
$\boldsymbol{D}_i$	Fourth order elasticity tensor associated with i-th grain
e_j	Set of nodes enriched by the j-th enrichment function
E	Young's modulus
$f(\cdot)$	Linear functional
$\boldsymbol{F}$	Force vector
$\boldsymbol{g}_i$	Integration points
h	Discretization parameter
H_{Γ_C}	Heaviside function associated with crack Γ_C

$H^1(\Omega)$	Subspace of functions in $L^2(\Omega)$ whose weak partial derivatives are in $L^2(\Omega)$
$H^2(\Omega)$	Subspace of functions in $L^2(\Omega)$ whose weak partial derivatives up to an order of 2 are in $L^2(\Omega)$
$H^1_{\Gamma_D}(\Omega)$	Subspace of functions in $H^1(\Omega)$ that vanish at Γ_D
H	Hilbert space
$\boldsymbol{H}$	Matrix used to describe the numerical enrichment procedure
i,j,k,l	Indices
$\boldsymbol{I}$	Identity
$\boldsymbol{I}_{\Omega_i}$	Indicator function
$\boldsymbol{J}$	Coordinates of a junction
$\boldsymbol{K}$	Stiffness matrix
$\mathcal{K}(\boldsymbol{K}, \boldsymbol{f})^i$	i-th Krylov space of $\boldsymbol{K}$ and $\boldsymbol{f}$
l	Rate of convergence
N_i	Piecewise polynomial ansatz functions for the FEM
N_f,N,N^i_{min}	Number of temperature cycles to failure
$\boldsymbol{n}$	Normal vector
$p_0, ..., p_i$	Coefficients of a polynomial of order i
$P(\boldsymbol{x})$	Weakly singular enrichment function
P^i	Space of polynomials of order i
$\boldsymbol{P}$	Preconditioner
q_R	Rayleigh quotient
$\boldsymbol{Q}$	Orthogonal matrix or rotation
r	Radial component of polar coordinates
r_{max}	Enrichment radius
r_j	Set of nodes forming the ramp function of the j-th enrichment function

$R(\boldsymbol{x})$	Ramp function
R_A	Range of the operator A
$\boldsymbol{r}_i$	Residual in the i-th step of the MINRES-solver
$\mathbb{R}^n$	Euclidean space of dimension n
$s_{rr}, s_{r\theta}, ...$	Auxiliary variables used to derive the stress-strain relationship in cylindrical coordinates
S	Subspace
S^h	Subspace arising from some sort of discretization
$S^\perp$	Space orthogonal to S
t	Time
T	Temperature
T_{ref}	Reference temperature
$\boldsymbol{t}_i$	Vector forming the matrix $\boldsymbol{T}_i$
$\boldsymbol{T}_i$	Tridiagonal matrix in the i-th step of the MINRES-solver
$\boldsymbol{t}$	Cauchy stress vectors or traction surface forces respectively
u, v, w	Scalar functions
$\boldsymbol{u}$, $\boldsymbol{v}$	Vector fields describing the displacement or the velocity of a structure
$\boldsymbol{u}^h$, $\boldsymbol{v}^h$	Elements of S^h
$\boldsymbol{u}^1, ..., \boldsymbol{u}^n$	Basis for S^h
w_i	Integration weights
$\boldsymbol{x}$	Coordinate vector in $\mathbb{R}^3$
$\boldsymbol{x}_i$	Location of the i-th node
$\boldsymbol{Y}$	Matrix used for preconditioning
$\boldsymbol{Z}$	Penalty vectors
da	Infinitesimal area

dv	Infinitesimal volume
dt	Infinitesimal time step
$DOFs$	Degrees of freedom
$\boldsymbol{\alpha}$	Thermal expansion coefficients
Γ	Boundary of the open domain Ω
Γ_D	Part of the boundary at which Dirichlet boundary conditions are prescribed
Γ_N	Part of the boundary at which Neumann boundary conditions are prescribed
Γ_B	Grain boundary segment
Γ_C	Crack segment
ϵ	Small value tending to zero or scalar strains
$\boldsymbol{\epsilon}$	Infinitesimal strain tensor
$\boldsymbol{\zeta}$	Variable vector used in the numerical enrichment procedure
η, η^h	Penalty terms
θ	Angular component of polar coordinates
$\kappa(\cdot)$	Condition number of a matrix
λ	Eigenvalue
ν	Poisson's ratio
ξ	Auxiliary variable introduced to derive the stress-strain relationship in cylindrical coordinates
Π	Projector
ρ	Mass density
σ_{vm}	von Mises stresses
$\boldsymbol{\sigma}$	Stress tensor
ϕ	Enrichment function
χ	Cutoff function

ψ	Angular component in $P(\boldsymbol{x})$
Ω	Structural domain, open subset of $\mathbb{R}^2$ or $\mathbb{R}^3$
Ω_i	i-th grain

1. Introduction

1.1. Numerical simulation in industrial applications and the extended finite element method

Numerical simulation has an increasing impact in many areas of science and engineering. The ability to make predictions about the behavior of physical systems brings several advantages. In industrial applications the performance of a product can be addressed before it is actually built. If planes or ships are considered it is absolutely crucial to decide whether a certain design performs well because of the immense costs associated with its fabrication.

But also for smaller products numerical simulation can bring significant advantages to a company. In order to guarantee a certain quality, the lifetime in the field must be estimated. A possible approach would be to expose the products to experimental conditions that are close to the ones in the field. In that case the expected lifetime is a direct result of the experiments.

But some products have a life expectancy of several years. Conducting the corresponding experiments would be very time-consuming. In that case accelerated reliability tests can be performed. The conditions are much harsher in these tests than the ones in the field, which causes the product to fail earlier. The expected lifetime in the field is obtained by extrapolation.

But even those tests might take too much time. The production of test samples involves a lot of planning and the tests can only be accelerated by a certain factor, otherwise the failure mode might change and extrapolation

to field conditions is meaningless. Therefore, numerical simulation is often combined with experimental testing: Once a certain numerical method for lifetime prediction is validated experimentally, it can be used for similar designs and testing conditions as well.

Lifetime prediction by numerical simulation is used to decide whether a product design should be produced or not, but also in the process of design optimization. In order to find a reasonable balance between production costs and quality, repeated design changes have to be performed and evaluated. The quality of an altered design in terms of lifetime is evaluated by numerical simulation. Since the results are usually available in a shorter amount of time, the design process becomes much more efficient.

In order to make predictions about a physical system a model is needed. The true behavior of the system is approximated by a mathematical description. Partial differential equations (PDEs) are often employed. The analytical solution of general PDEs is only possible in special cases. Numerical methods, however, make it possible to determine approximations to the exact solution for a much greater class of problems. The accuracy and the efficiency of these approximations depend on the method itself and how it is applied.

This thesis will only deal with physical problems that can be described in the framework of continuum mechanics. In this context the method that is most widely used is the finite element method (FEM). An overview of its basic features is given in section 2.3. Often the displacements of the structure in question are the primary unknowns. The approximation space is chosen implicitly by defining a mesh which covers the structural domain. For two-dimensional structures the mesh may for example consist of triangles and quadrilaterals. The triangles and quadrilaterals are called elements. The possible approximations are all functions that are continuous throughout the structural domain and formed by polynomials of some order inside each element. The quality of the approximation is mainly determined by the ability of this function space to reproduce the exact solution. The approximation error can be decreased by refining the mesh. Small elements, however, increase the computational effort because the overall number of elements is higher and thus the dimension of the corresponding equation systems. In some situations even a very fine mesh produces unacceptable

solutions, because the space of piecewise polynomial functions is not appropriate to approximate certain characteristic features of the exact solution. The idea of enrichment is to add functions to the function space which are able to model these features. By understanding the physics, special enrichment functions can be tailored for a particular problem. The extended finite element method (X-FEM) provides a framework for introducing these functions into the FEM function space. Often the approximation properties of the FEM function space can be enhanced significantly. The X-FEM was developed mainly during the last ten years. There are only a few commercial implementations and the method is an active field of research. A review of the current status of the method is given in section 2.4.
X-FEM is especially useful if complicated geometries are considered. To obtain acceptable solutions via FEM, the element edges have to align with geometrical features such as the boundary of the structure, material interfaces and cracks. Additionally, the meshes must be particularly fine in the vicinity of crack tips and re-entrant corners to resolve the high strain gradients appearing in these areas. Automatic meshing algorithms can be used to generate meshes that fulfill these geometric conditions while maintaining a good mesh quality. But there is no guarantee that these algorithms generate any mesh at all. The more complicated the geometry becomes, the more likely it is that the meshing algorithms fail. Using X-FEM the enrichment functions can be chosen to represent the geometrical features mentioned above. Very simple meshes that are independent of the geometry may therefore be used in combination with the X-FEM to avoid problems during mesh generation.

1.2. Lifetime prediction for solder joints

This thesis is a joint project between the Robert Bosch GmbH and the University of Glasgow. Next to a series of other products, Bosch produces electronic devices for cars. In the application part of this thesis we will therefore be concerned with the reliability of automotive electronics. To produce these devices, electric components such as resistors, chips etc. are soldered onto a circuit board. The solder joints form a mechanical as well

as an electrical connection. The lifetime of these devices is determined by many factors. The most important damage mechanism is thermomechanical fatigue.

If the car is used, the motor heats up. Elevated temperatures in the vicinity of the motor cause the different components of the device to expand. The electronic components expand much less than the circuit board. This generates an expansion mismatch. Since the solder joints connect both, they have to accommodate this mismatch. Stresses are generated inside the joint and plastic deformation occurs. After the motor is turned off, the temperature decreases. This causes the solder joints to deform in the opposite direction. The repeated plastic deformation causes cracks to initiate and to propagate. Once the crack propagated through the whole joint the device fails. A short discussion of solder joints in electronic devices and thermomechanical fatigue is given in section 4.1.

Automotive electronics are expected to last several years. Performing actual field tests is therefore not feasible. Accelerated tests still may take several months. This constitutes the necessity for a lifetime prediction methodology based on simulation.

The lifetime prediction methodology currently used at Bosch and other companies follows a simple scheme. All the materials of an electronic device are characterized by experimentally determined material parameters and a continuum mechanical description involving elastic and creep deformation coupled with thermal expansion. The solder joint is assumed to be a homogeneous isotropic material without cracks or damage. The creep that accumulates over time is extracted from the FEM simulation and correlated with the experimentally determined lifetime for a given temperature profile by fitting constants in a statistical approach called the Coffin-Manson equation.

For slight changes in the temperature profile or the geometry the constants in this equation might still be valid and the lifetime can be predicted without conducting new experiments. But for greater changes in design, temperature profile etc. new experiments have to be conducted. This results in a large database of Coffin-Manson constants. Such a database has been developed for the lead-containing solder SnPb over the last 50 years. But due to environmental concerns and health issues the European Union has banned

lead-containing solders. SnAgCu has replaced SnPb in most applications. However, for SnAgCu this database is missing. Instead of developing a new database in numerous expensive and time-consuming experiments, the problem could be resolved by using more advanced simulation techniques.
The main problem with the current methodology is that the actual damage mechanisms are not included in the model and therefore it is not realistic. Only a few attempts have been made to simulate the damage in solder joints. A review of the literature available on the topic of lifetime prediction for solder joints is given in section 4.2.
The discussion in section 4.3 constitutes that the damage in solder joints is closely related to the microstructure. The joint is formed by several crystals. Areas in which the same crystal orientation can be observed are called grains. Most of them are relatively large compared to the joint measurements. The crystals have an anisotropic elastic behavior. Their stiffness can vary up to a factor of three depending on the orientation. Because the plastic deformation and hence also the damage mechanism depend on the elastic deformation, the crystal orientations have a large impact on the lifetime of the joints. Thus, to model the damage mechanisms accurately, the polycrystalline structure has to be a part of the solder model.
In this thesis a new lifetime prediction methodology is proposed based on a solder model that incorporates the microstructural properties of the joint. The novelties introduced to the field of solder joint lifetime prediction by this methodology can be summarized as follows:

- In section 4.3 it is explained how random computer models of realistic solder joint microstructures can be generated artificially. A procedure to calculate crack growth for these structures is explained. This is the first model that couples crack growth in solder joints with realistic microstructures.

- To describe the different grains forming the joint mechanically, constitutive laws for the grains are determined in section 4.4.1 by an inverse procedure.

- To capture the variety of possible microstructures in electronic devices it is proposed to generate a series of random microstructures. By performing crack growth calculations for each of these structures,

the influence of varying microstructures on the reliability can be addressed. In contrast to the classical semi-empirical methods which only predict an average lifetime, much more reliable conclusions could be drawn from such an approach in industrial applications. Numerical examples are given in section 4.4.2. The results are compared with experimental data.

1.3. Development of X-FEM towards the simulation of complex polycrystalline structures

To realize the lifetime prediction methodology mentioned above it is crucial to automate the crack growth simulations. The main problem in automating these simulations is to guarantee a successful mesh generation. Due to the complexity of the randomly generated microstructures conventional mesh generators may fail. In that case user intervention is necessary. However, in an industrial application a large number of random microstructures must be generated to get a reliable failure probability. To use the lifetime prediction methodology efficiently it must therefore be guaranteed that mesh generation can be performed by an algorithm completely without user intervention.

The enrichment functions of the X-FEM enable engineers to use meshes which are less dependent on geometrical features. As a result the meshing procedure can be simplified and automated. X-FEM may be interpreted as a framework of several methods and ideas that are based on a common principle. Several approaches can be found in the literature that can simplify the task of simulating the complex microstructure of solder joints. What is missing is a technique to determine enrichment functions which represent general strain singularities in linear elastic polycrystalline structures. These singularities can occur at junctions formed by several grain boundaries, at crack tips or at re-entrant corners.

The corresponding enrichments would help to realize a fully automated crack growth calculation in solder joints. Their shape is dependent on the orientation of the surrounding grains and their shape. Due to the large

number of possible microstructural configurations it is not possible to write these functions in a closed form.
Another problem which occurs during the simulation of microstructures with the X-FEM is the ill-conditioning of the stiffness matrix. Ill-conditioned stiffness matrices can slow down the solution process significantly. In the FEM several criteria for evaluating the mesh quality exist, which guarantee well-conditioned stiffness matrices. For the X-FEM such criteria do not exist. In fact in section 3.3.1 it will be shown that currently used enrichment schemes can result in arbitrarily ill-conditioned matrices in special cases.
In order to make the X-FEM applicable to the lifetime prediction methodology the following ideas and novelties are put forward in this thesis:

- A numerical procedure to determine enrichment functions that represent arbitrary strain singularities in polycrystalline structures described by plane strain linear elasticity is explained and tested in section 3.2. The enrichments have been combined with other currently used enrichment types (cf. section 2.4) to arrive at a method which is able to restore optimal convergence rates in the presence of weak singularities.

- An efficient preconditioning technique which guarantees well-conditioned stiffness matrices with the X-FEM and arbitrary enrichments is introduced in section 3.3.

- A meshing strategy for the X-FEM simulation of grain structures generated by a Voronoi-tessellation is introduced in 3.4. Due to its simplicity the strategy can be automated completely. Meshing has to be performed only once at the beginning of a crack growth calculation. The growing crack is accounted for by introducing new enrichments.

- This is the first application of X-FEM to thermomechanical fatigue of solder joints.

These features have been implemented in a Matlab code. To simulate the transient behavior of solder joints undergoing thermal cycling, an approach to calculate thermal expansion and creep deformation has additionally been implemented. A penalty method was used to prevent crack faces from over-

lapping. The final version of the code was used to calculate the examples in section 4.4.2.

2. Theoretical background

This chapter aims to provide the theoretical background for the methods and ideas discussed in the other chapters. In section 2.1 the equations necessary to describe the structural problems considered in this thesis in the framework of continuum mechanics are listed and explained. In section 2.2 a brief introduction to the mathematical background is given. In section 2.3 the FEM and its approximation error is discussed. This will help us to understand the idea of the X-FEM and its enrichment functions as introduced in section 2.4. The MINRES iterative solver and its numerical properties are discussed in section 2.5 since this solver will be used to solve the equation systems in this thesis. Furthermore the relationship between the condition number of an equation system and the number of iterations will be discussed. This will help us to understand the preconditioning technique developed in section 3.3.

2.1. Material description and constitutive relations

The equations introduced in this section can be derived in the framework of continuum mechanics. However, this will be omitted here, the interested reader is referred to the work of Liu [1] and Braess [2].
If $\boldsymbol{u}$ is a vector field describing the displacements of a structure $\boldsymbol{\Omega} \subset \mathbb{R}^3$, we define the linearized strain tensor by:

$$\boldsymbol{\epsilon} := \frac{1}{2}(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T) \tag{2.1}$$

Using $\boldsymbol{\epsilon}$ to develop constitutive laws is appropriate if the displacements considered are small.
We will be concerned with three different phenomena materials can exhibit: Elasticity, creep and thermal expansion.
Elastic behavior can be characterized as follows. Assume that a load is applied to the structure. In a purely elastic material, if the load is reversed, the structure returns to its original state. Physically this means that the atomic structure is deformed but not rearranged.
This is different if a structure shows creep behavior. Creep is a form of plasticity. Plastic behavior is physically characterized by a rearrangement of the atomic structure. The term creep is used for time-dependent plastic processes. Because the atomic structure is rearranged during creep, deformation energy is transformed to heat. Therefore, the structure does not return to its original state if an applied load is reversed.
Thermal expansion of the structure happens due to an increase in temperature. The temperature in a structure is roughly speaking the average vibration of the atoms making up the structure. If the kinetic energy of the atoms increases, this causes the structure macroscopically to expand.
We assume that all the deformations that have just been described can be superimposed to obtain $\boldsymbol{\epsilon}$:

$$\boldsymbol{\epsilon} = \boldsymbol{\epsilon}^{el} + \boldsymbol{\epsilon}^{cr} + \boldsymbol{\epsilon}^{th} \tag{2.2}$$

$\boldsymbol{\epsilon}^{el}$ are the elastic strains, $\boldsymbol{\epsilon}^{cr}$ are the creep strains and $\boldsymbol{\epsilon}^{th}$ are the thermal strains.
The problems considered in this thesis are either static problems or time-dependent problems in which the structure deforms slowly. In the latter case the structure can be modeled as quasi-static (i.e. no inertia effects). Therefore, the equation of motion becomes:

$$\mathrm{div}(\boldsymbol{\sigma}) + \boldsymbol{b} = 0 \tag{2.3}$$

$\boldsymbol{\sigma}$ is the symmetric stress tensor and $\boldsymbol{b}$ is a body force distribution.
A relation between $\boldsymbol{\sigma}$ and the elastic strains $\boldsymbol{\epsilon}^{el}$ is given by the fourth-order stiffness tensor $\boldsymbol{D}$:

$$\boldsymbol{\sigma}_{kl} = \boldsymbol{D}_{ijkl}\boldsymbol{\epsilon}^{el}_{ij} \tag{2.4}$$

Summation over repeated indices is understood here. The stress tensor must be symmetric such that conservation of angular momentum is fulfilled, $\boldsymbol{\epsilon}^{el}$ is symmetric by definition and therefore $\boldsymbol{D}$ must also obey certain symmetry relations:

$$\boldsymbol{D}_{ijkl} = \boldsymbol{D}_{jikl} \tag{2.5}$$

$$\boldsymbol{D}_{ijkl} = \boldsymbol{D}_{ijlk} \tag{2.6}$$

$$\boldsymbol{D}_{ijkl} = \boldsymbol{D}_{klij} \tag{2.7}$$

This leaves a total of 21 independent components of $\boldsymbol{D}$.
Creep deformation will be modeled by:

$$\boldsymbol{\epsilon}^{cr} = \int_0^t \frac{\partial \boldsymbol{\epsilon}^{cr}}{\partial t} dt \tag{2.8}$$

$$\frac{\partial \boldsymbol{\epsilon}^{cr}}{\partial t} = \begin{cases} 0 & \text{if } \sigma_{vm} = 0 \\ f(\sigma_{vm}, T)\frac{\boldsymbol{\sigma}'}{\sigma_{vm}} & \text{otherwise} \end{cases} \tag{2.9}$$

The function $f(\sigma_{vm}, T)$ is a nonnegative smooth scalar function determining the creep rate. $f(\sigma_{vm}, T)$ is assumed to be zero if $\sigma_{vm} = 0$. Furthermore, it should be a monotonically increasing function of T and σ_{vm}. σ_{vm} are the von Mises stresses and $\boldsymbol{\sigma}'$ is called the stress deviator. The stress deviator is defined by:

$$\boldsymbol{\sigma}' := \boldsymbol{\sigma} - \frac{\sigma_{11} + \sigma_{22} + \sigma_{33}}{3}\boldsymbol{I} \tag{2.10}$$

If σ_1, σ_2 and σ_3 are the three eigenvalues of the stress tensor, then the von Mises stresses can be defined by:

$$\sigma_{vm} := \sqrt{\frac{1}{2}\left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right]} \tag{2.11}$$

Thermal expansion will be modeled by:

$$\boldsymbol{\epsilon}^{th} = \boldsymbol{\alpha}(T - T_{ref}) \tag{2.12}$$

T_{ref} is a reference temperature and $\boldsymbol{\alpha}$ is the symmetric tensor of thermal coefficients chosen such that $\boldsymbol{\epsilon}^{th}$ are the strains measured due to a temperature increase of $(T - T_{ref})$.

2.2. Functional analysis

To simplify the discussion we will assume homogeneous displacement boundary conditions. We will use Γ to label the boundary of a domain $\Omega \subset \mathbb{R}^3$. The boundary is divided into two parts Γ_N and Γ_D. Zero displacements are prescribed at Γ_D and a traction $\boldsymbol{t}$ is applied to the structure at the boundary Γ_N with $\boldsymbol{n}$ being the unit normal along Γ_N.
The boundary value problem then becomes:

$$\operatorname{div}(\boldsymbol{\sigma}) + \boldsymbol{b} = 0 \quad \text{if } \boldsymbol{x} \in \Omega \tag{2.13}$$

$$\boldsymbol{\sigma n} = \boldsymbol{t} \quad \text{if } \boldsymbol{x} \in \Gamma_N \tag{2.14}$$

$$\boldsymbol{u} = \boldsymbol{0} \quad \text{if } \boldsymbol{x} \in \Gamma_D \tag{2.15}$$

The relationship between stresses, strains and displacements is given by (2.1) and (2.4).
To reformulate the problem we multiply equation (2.13) with a vector-valued smooth function $\boldsymbol{v}^T$ that vanishes at Γ_D. Integrating the new expression over the whole domain results in:

$$\int_\Omega \boldsymbol{v}^T(\nabla\boldsymbol{\sigma}(\boldsymbol{u}))dv + \int_\Omega \boldsymbol{v}^T\boldsymbol{b}dv = 0 \tag{2.16}$$

Applying integration by parts to the first term and exploiting the symmetry of the stress tensor gives:

$$\int_\Omega \boldsymbol{\epsilon}(\boldsymbol{v}) : \boldsymbol{\sigma}(\boldsymbol{u})dv = \int_\Omega \boldsymbol{v}^T\boldsymbol{b}dv + \int_{\Gamma_N} \boldsymbol{v}^T\boldsymbol{t}da \tag{2.17}$$

The weak formulation of the problem is known as:

Find $\boldsymbol{u}$ such that (2.17) holds for all (sufficiently smooth) functions $\boldsymbol{v}$.

Equation (2.13) is called the strong form. Obviously a solution to the strong problem formulation (which also fulfills the boundary conditions) is a solution to (2.17) for any (sufficiently smooth) $\boldsymbol{v}$. But on the other hand if a function $\boldsymbol{u}$ solves (2.17) for every $\boldsymbol{v}$, it is not necessarily a solution of the strong problem formulation.
To address the existence and the uniqueness of the solution of the weak formulation, we need to work in the appropriate function spaces. But first

some tools from functional analysis are needed. For a good introduction to this topic the book by Kreyszig [3] could be considered.

Theorem 2.1 (Projection theorem) *Let H be a Hilbert space and S a non-empty closed subspace. Then for any $u \in H$ there exist a unique element $v \in S$ such that:*

$$\|u - v\| = \inf_{w \in S} \|w - u\| \tag{2.18}$$

Proof cf. [4]. □

In other words for every $u \in H$ there is a unique element $v \in S$ which is closest to u in the induced norm. This is also called the projection of u onto S. This theorem makes it possible to use the notion of orthogonality in general Hilbert spaces. The fact that Hilbert spaces are by definition complete is important since generalized projections are defined by a limit process.
Using the projection theorem one can make a very powerful statement about bounded linear functionals on Hilbert spaces. A bounded linear functional $f(\cdot)$ on a Hilbert space H is a linear mapping $H \to \mathbb{R}$ such that there is a positive constant c with $f(u) \leq c\|u\|$ for all $u \in H$. It is worth mentioning that the boundedness of a linear functional is equivalent to its continuity [4]. This fact is used in the proof of the following theorem.

Theorem 2.2 (Riesz representation theorem) *Let H be a Hilbert space and $f(\cdot)$ a bounded linear functional. Then there exists a unique element $u \in H$ such that:*

$$f(v) = \langle u, v \rangle \quad \forall v \in H \tag{2.19}$$

Proof If $f = 0$ for all $v \in H$ choose $u = 0$. So let us assume $f \neq 0$ for some v. Then we define:

$$S := \{v \in H | f(v) = 0\} \tag{2.20}$$

At least 0 is in S.
S is a subspace of H, for $v_1, v_2 \in S$ and $\lambda \in \mathbb{R}$:

$$f(v_1) + f(v_2) = f(v_1 + v_2) = 0 \tag{2.21}$$

$$\lambda f(v_1) = f(\lambda v_1) = 0 \tag{2.22}$$

Because boundedness of the functional implies continuity, the space S is closed and thus complete: A Cauchy sequence in S has a limit in H, due to the continuity of f the function value of this limit will also be zero. Thus, the limit is in S.
Therefore, we can define the orthogonal space $S^{\perp}$. By assumption this space does not only contain 0. Pick an arbitrary $\tilde{u} \neq 0$ and set:

$$u := \frac{\tilde{u} f(\tilde{u})}{\|\tilde{u}\|^2} \tag{2.23}$$

Then:

$$\langle u, v \rangle = 0 = f(v) \quad \forall v \in X \tag{2.24}$$

$$\langle u, v \rangle = f(v) \quad \forall v \in \text{span}(\tilde{u}) \tag{2.25}$$

It remains to show that any $v \in S^{\perp}$ is in $\text{span}(\tilde{u})$.
So assume that there are two linearly independent (non-zero) elements u_1 and u_2 in $S^{\perp}$. Then:

$$0 = \frac{f(u_1)}{f(u_1)} - \frac{f(u_2)}{f(u_2)} = f(\frac{u_1}{f(u_1)} - \frac{u_2}{f(u_2)}) \tag{2.26}$$

Thus, a linear combination of u_1 and u_2 is an element of S, which leads to a contradiction. □

We can define variational problems on Hilbert spaces using bilinear forms. A bounded coercive symmetric bilinear form defined on a Hilbert space H is a mapping $a(\cdot, \cdot) : H \times H \to \mathbb{R}$ with:

$$1) \quad a(u, v) = a(v, u) \quad \forall u, v \in H \quad \text{(Symmetry)} \tag{2.27}$$

$$2) \quad \exists C > 0 \mid a(u, v) \leq C\|u\|\|v\| \quad \forall u, v \in H \quad \text{(Boundedness)} \tag{2.28}$$

$$3) \quad \exists c > 0 \mid a(v, v) \geq c\|v\|^2 \quad \forall v \in H \quad \text{(Coercivity)} \tag{2.29}$$

The next theorem proves existence and uniqueness for variational problems on general Hilbert spaces:

Theorem 2.3 (Lax-Milgram Lemma) *Let H be a Hilbert space, $a(\cdot, \cdot)$ a bounded coercive symmetric bilinear form and $f(\cdot)$ a bounded linear func-*

tional. Then there exists a unique element $u \in H$ such that:

$$a(u,v) = f(v) \quad \forall v \in H \tag{2.30}$$

Proof From the Riesz representation theorem it follows due to the boundedness of $a(\cdot,\cdot)$ that we have for some bounded linear operator $\mathcal{A}$:

$$a(u,v) = \langle \mathcal{A}u, v \rangle \tag{2.31}$$

Also for some $w \in H$:

$$f(v) = \langle w, v \rangle \tag{2.32}$$

We have to show that the operator $\mathcal{A}$ is a bijective mapping, then we can set $u := \mathcal{A}^{-1}w$ and the proof is finished. Assume $\mathcal{A}$ is not injective, then there are two distinct $u_1, u_2 \in H$ with $\mathcal{A}u_1 = \mathcal{A}u_2$. But then we get a contradiction to (2.29) by:

$$a(u_1 - u_2, u_1 - u_2) = \langle \mathcal{A}(u_1 - u_2), u_1 - u_2 \rangle = 0 \tag{2.33}$$

To show surjectivity we first have to show that the range $R_{\mathcal{A}}$ of $\mathcal{A}$ is closed. Assume that this is not true, then there exists a Cauchy sequence $w_i = Au_i$ such that u_i is not a Cauchy sequence (please note that boundedness of $\mathcal{A}$ implies that the limit of any sequence u_i is the limit of $\mathcal{A}u_i$). But then we have a sequence of elements $u_i - u_{i+1}$ whose norm does not tend to zero. Thus, we have:

$$\|\mathcal{A}(u_i - u_{i+1})\| = \underbrace{\|w_i - w_{i+1}\|}_{\to 0} \tag{2.34}$$

Then by using (2.31) we get a contradiction from:

$$\langle \mathcal{A}(u_i - u_{i+1}), u_i - u_{i+1} \rangle \geq c\|u_i - u_{i+1}\|^2 \tag{2.35}$$

$$\underbrace{\left\langle \mathcal{A}(u_i - u_{i+1}), \frac{u_i - u_{i+1}}{\|u_i - u_{i+1}\|} \right\rangle}_{\to 0} \geq \underbrace{c\|u_i - u_{i+1}\|}_{>0} \tag{2.36}$$

The fact that the left-hand side vanishes is a consequence of the Cauchy-Schwarz inequality.
Therefore, we are allowed to construct $R_{\mathcal{A}}^{\perp}$. If $R_{\mathcal{A}}^{\perp} = \{0\}$ we have shown

surjectivity. Assume that this is not true, then there is a non-zero $v \in R_{\mathcal{A}}^{\perp}$. But then we get a contradiction to (2.29) by:

$$\langle \mathcal{A}v, v \rangle = a(v, v) = 0 \tag{2.37}$$

□

Therefore, we can show existence and uniqueness of a solution of the structural problem in its weak form if we can formulate it in terms of a bilinear form $a(\cdot, \cdot)$ and a linear functional $f(\cdot)$ such that the assumptions in theorem 2.3 are fulfilled.
The idea is to interpret the solution of the structural problem as a vector in a Hilbert space. An appropriate Hilbert space whose elements are scalar functions on Ω, is the Sobolev space $H^1(\Omega)$. For a good introduction to these spaces the reader is referred to the book by Alt [4]. The space $H^1_{\Gamma_D}(\Omega)$ contains the subset of functions which vanish in the vicinity of Γ_D. Vector-valued functions can be constructed by using the product space:

$$\left(H^1_{\Gamma_D}(\Omega)\right)^3 := H^1_{\Gamma_D}(\Omega) \times H^1_{\Gamma_D}(\Omega) \times H^1_{\Gamma_D}(\Omega) \tag{2.38}$$

$\left(H^1_{\Gamma_D}(\Omega)\right)^3$ is again a Hilbert space. Using the body forces $\boldsymbol{b}$ from (2.17) we set:

$$\hat{\boldsymbol{b}} := \boldsymbol{D} : \left(\boldsymbol{\epsilon}^{th} + \boldsymbol{\epsilon}^{cr}\right) + \boldsymbol{b} \tag{2.39}$$

We define the linear functional as:

$$f(\boldsymbol{v}) := \int_{\Omega} \boldsymbol{v}^T \hat{\boldsymbol{b}} dv + \int_{\Gamma_N} \boldsymbol{v}^T \boldsymbol{t} da \tag{2.40}$$

Using equation (2.2) and equation (2.4) we see that (2.17) can be written in the form of equation (2.30) where $f(\cdot)$ is defined by (2.40) and the bilinear form is defined by:

$$a(\boldsymbol{u}, \boldsymbol{v}) := \int_{\Omega} \boldsymbol{\epsilon}(\boldsymbol{u}) : \boldsymbol{D} : \boldsymbol{\epsilon}(\boldsymbol{v}) dv \tag{2.41}$$

Using equations (2.41) and (2.40) we can now restate the weak problem as:

Find $\boldsymbol{u} \in \left(H^1_{\Gamma_D}(\Omega)\right)^3$ such that $a(\boldsymbol{u}, \boldsymbol{v}) = f(\boldsymbol{v})$ holds for all $\boldsymbol{v} \in \left(H^1_{\Gamma_D}(\Omega)\right)^3$.

Due to the definition of the Sobolev spaces this only makes sense if the integration in (2.41) and (2.40) is interpreted as the Lebesgue integration and the differentiation is interpreted as the weak differentiation.
One can show that the bilinear form in (2.41) fulfills all necessary conditions if Γ_D has a positive measure, that is, all rigid body motions are suppressed [2]. In that case, assuming that $f(\boldsymbol{v})$ is bounded, the weak problem formulation has a unique solution.

2.3. Finite element method

To solve the weak formulation of our structural problem numerically we consider finite-dimensional subspaces S^h of $(H^1_{\Gamma_D}(\Omega))^3$. Since finite dimensional subspaces of Hilbert spaces are again Hilbert spaces [3], the weak formulation can be solved uniquely in this subspace too. Using (2.41) and (2.40) the weak formulation becomes:

$$\text{Find } \boldsymbol{u}^h \in S^h \text{ such that for all } \boldsymbol{v}^h \in S^h\text{:}$$

$$a(\boldsymbol{v}^h, \boldsymbol{u}^h) = f(\boldsymbol{v}^h) \tag{2.42}$$

If $\boldsymbol{u}_1, ..., \boldsymbol{u}_n$ is a basis of S^h the element $\boldsymbol{u}^h$ can be written as:

$$\boldsymbol{u}^h = a_1\boldsymbol{u}_1 + ... + a_n\boldsymbol{u}_n \tag{2.43}$$

To determine the unknown coefficients $a_1, ..., a_n$ we insert (2.43) in (2.42) and postulate that the equation holds for all basis functions:

$$\begin{aligned} a(\boldsymbol{u}^1, a_1\boldsymbol{u}^1 + ... + a_n\boldsymbol{u}^n) &= f(\boldsymbol{u}^1) \\ &\vdots \\ a(\boldsymbol{u}^n, a_1\boldsymbol{u}^1 + ... + a_n\boldsymbol{u}^n) &= f(\boldsymbol{u}^n) \end{aligned} \tag{2.44}$$

If these equations hold, then certainly $a(\boldsymbol{v}^h, a_1\boldsymbol{u}^1 + ... + a_n\boldsymbol{u}^n) = f(\boldsymbol{v}^h)$ holds for every $\boldsymbol{v}^h \in S^h$.

Due to the linearity of $a(\cdot,\cdot)$ we can rewrite (2.44) as:

$$\begin{bmatrix} a(\boldsymbol{u}^1,\boldsymbol{u}^1) & \dots & a(\boldsymbol{u}^1,\boldsymbol{u}^n) \\ & \vdots & \\ a(\boldsymbol{u}^n,\boldsymbol{u}^1) & \dots & a(\boldsymbol{u}^n,\boldsymbol{u}^n) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f(\boldsymbol{u}^1) \\ \vdots \\ f(\boldsymbol{u}^n) \end{bmatrix} \tag{2.45}$$

Therefore, to determine the unknown coefficients one has to solve a linear equation system:

$$\boldsymbol{K}\boldsymbol{a} = \boldsymbol{f} \tag{2.46}$$

If the functions $u^1, ..., u^n$ are linearly independent, the equation system has a unique solution.
This is also known as the Ritz-Galerkin method. The matrix $\boldsymbol{K}$ is called the stiffness matrix. The question how the result is related to the exact solution immediately arises. The answer is given by the following theorem.

Theorem 2.4 (Cea's Lemma) *Let H be a Hilbert space and S^h a finite-dimensional subspace. Let $a(\cdot,\cdot)$ be a bounded, coercive, symmetric bilinear form and $f(\cdot)$ a bounded linear functional defined on H. The solution $\boldsymbol{u}$ of the corresponding variational problem on H is related to the solution $\boldsymbol{u}^h$ on S^h by:*

$$\|\boldsymbol{u} - \boldsymbol{u}^h\| \leq \frac{C}{c} \inf_{\boldsymbol{v}^h \in S^h} \|\boldsymbol{u} - \boldsymbol{v}^h\| \tag{2.47}$$

C and c are the constants in equations (2.29) and (2.28).

Proof For any $\boldsymbol{v}^h \in S^h$ we have:

$$a(\boldsymbol{u} - \boldsymbol{u}^h, \boldsymbol{v}^h) = a(\boldsymbol{u}, \boldsymbol{v}^h) - a(\boldsymbol{u}^h, \boldsymbol{v}^h) = f(\boldsymbol{v}^h) - f(\boldsymbol{v}^h) = 0 \tag{2.48}$$

Since $\boldsymbol{u}^h - \boldsymbol{v}^h \in S^h$ and by the definition of $a(\cdot,\cdot)$ we obtain:

$$\begin{aligned} & c\|\boldsymbol{u} - \boldsymbol{u}^h\|^2 && (2.49)\\ \leq & a(\boldsymbol{u} - \boldsymbol{u}^h, \boldsymbol{u} - \boldsymbol{u}^h) && (2.50)\\ = & a(\boldsymbol{u} - \boldsymbol{u}^h, \boldsymbol{u} - \boldsymbol{u}^h) + a(\boldsymbol{u}^h - \boldsymbol{v}^h, \boldsymbol{u} - \boldsymbol{u}^h) && (2.51)\\ = & a(\boldsymbol{u} - \boldsymbol{v}^h, \boldsymbol{u} - \boldsymbol{u}^h) && (2.52)\\ \leq & C\|\boldsymbol{u} - \boldsymbol{v}^h\|\|\boldsymbol{u} - \boldsymbol{u}^h\| && (2.53) \end{aligned}$$

Division by $c\|\boldsymbol{u} - \boldsymbol{u}^h\|$ proves the lemma. □

Thus, the approximation error of $\boldsymbol{u}^h$ obtained by the Ritz-Galerkin method is bounded by the approximation error of the best possible solution in S_h. Another interpretation is possible. $a(\cdot,\cdot)$ defines an inner product on $(H^1_{\Gamma_D}(\Omega))^3$ if $\text{meas}(\Gamma_D) > 0$. In the proof it is shown that the approximation error is orthogonal to S_h with respect to $a(\cdot,\cdot)$. Therefore, the Ritz-Galerkin approximation minimizes the error in the norm induced by $a(\cdot,\cdot)$. This norm is also called the energy norm.

Different choices for S^h are possible. One could for instance choose polynomials up to a certain order. However, this is not advantageous from a computational point of view. If polynomials are used the entries in $\boldsymbol{K}$ will generally be non-zero. Therefore, the memory requirements grow quadratically with n. It is much easier to deal with sparse matrices, in which most of the entries are zero. Zero entries do not have to be stored, for sparse matrices it suffices to store the indices of the non-zero entries together with its numerical value.

Sparse matrices are also advantageous when iterative methods are used to solve the equation systems. Iterative solvers will be explored in section 2.5. An iterative solver multiplies in every step a vector with the matrix $\boldsymbol{K}$. Zero elements can be neglected in this multiplication. The computation time for one solution step of the solver therefore depends mainly on the number of non-zero entries.

Finite element methods are a subclass of Ritz-Galerkin methods. They are characterized by a special choice of the space S^h and its basis. The basis functions result in sparse matrices, and therefore the FEM has all the advantages mentioned above. Furthermore, the integration that is necessary to evaluate the matrix $\boldsymbol{K}$ can be done in a very simple and effective manner. The function space S^h is constructed implicitly by covering the structure with a mesh of polygonal elements such as triangles and quadrilaterals. A two-dimensional example of such a mesh formed by triangles is shown in figure 2.1. The points at which the element edges terminate are called nodes. The triangles have to obey certain rules: They have to be chosen such that no node is located at the interior of the edge of another element. Such a node would be called a hanging node. Furthermore, the triangles should cover the whole structural domain. Also the boundary Γ_D is formed by a

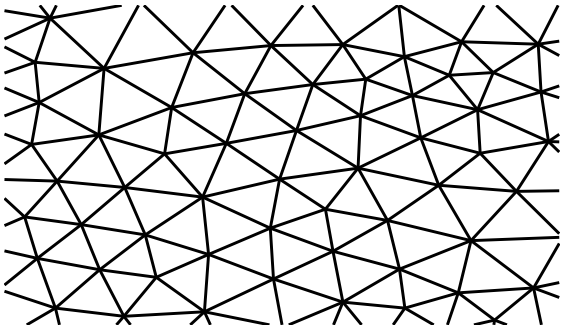


Figure 2.1.: *Example of a finite element mesh*

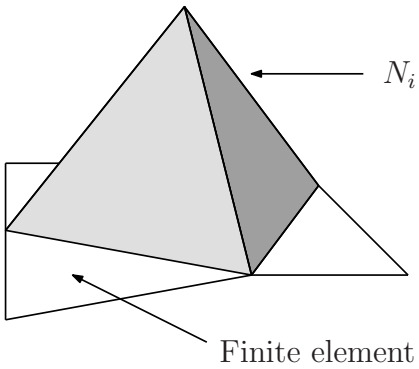


Figure 2.2.: *Visualization of the functions N_i*

set of exterior element edges. Only in that case the following construction of S^h is useful: Assuming that an index is assigned to each of the nodes which do not coincide with Γ_D, we define the functions $N_i(\boldsymbol{x})$ to be the functions which

- take the value 1 at the i-th node;
- take the value 0 at all other nodes;
- are a linear interpolation inside each element between the function values defined at the three nodes of that element.

An example of such a function is shown in figure 2.2. The space S^h is constructed using the following basis:

$$\boldsymbol{u}_1 = \begin{bmatrix} N_1(\boldsymbol{x}) \\ 0 \end{bmatrix} \ , \ \boldsymbol{u}_2 = \begin{bmatrix} 0 \\ N_1(\boldsymbol{x}) \end{bmatrix} \ , \ \boldsymbol{u}_3 = \begin{bmatrix} N_2(\boldsymbol{x}) \\ 0 \end{bmatrix} , ... \tag{2.54}$$

Other element choices are possible, although we will only be concerned with triangular elements in this thesis. The key feature here is that they are defined as piecewise polynomial functions, and that they vanish almost everywhere. This ensures that $a(\boldsymbol{u}_i, \boldsymbol{u}_j)$ is zero for almost all combinations of indices and therefore sparse matrices are obtained.
Instead of evaluating the stiffness matrix globally, one usually evaluates the stiffness matrix for each element and assembles the global matrix from these element stiffness matrices. This simplifies the algorithmic procedure and zero entries are automatically neglected.
Using piecewise polynomial functions also makes it possible to employ a simple integration scheme. Gauss integration of order one inside each element is sufficient to perform the integration of the element stiffness matrices exactly. The algorithm can further be simplified by using the well-known concept of isoparametric elements. The idea of isoparametric elements is to evaluate the derivative of the shape functions for each element on a common reference element. Thus, the shape of each element has not to be taken into account. The element stiffness matrices for the original elements are obtained by exploiting the properties of the corresponding integral transformation.
Theorem 2.4 is a rather abstract result. One is usually interested in the behaviour of the error during a sequence of mesh refinements. The convergence during mesh refinement is closely related to the regularity of the solution. We define:

$$\Pi u := \sum_i u(\boldsymbol{x}_i) N_i \tag{2.55}$$

Let h be the maximum length of the element edges, we are interested in how the error behaves if $h \to 0$. Convergence can usually only be guaranteed for meshes which are quasi-uniform. A series of meshes is called a quasi-uniform triangulation if the angles of all triangles are bounded from below by a positive constant.
A general statement about convergence properties is possible if the space $H^2(\Omega)$ is considered. This space is a subset of $H^1(\Omega)$ containing functions which have additional smoothness properties (cf. [4]). For instance these functions do not contain weak singularities.
Using the projector Π we can state the most important result about the

FEM-approximation for functions in $H^2(\Omega)$:

Theorem 2.5 *For a quasi-uniform triangulation and a function $u \in H^2(\Omega)$ there exists a constant c independent of h such that:*

$$\|u - \Pi u\|_{L^2} \leq ch^2 |u|_{H^2(\Omega)} \tag{2.56}$$

Proof cf. [5]. □

$\|\cdot\|_{L^2}$ is the L^2-norm and $|\cdot|_{H^2(\Omega)}$ is a semi-norm on $H^2(\Omega)$ [5]. Together with Cea's Lemma we get quadratic convergence in the L^2-norm for quasi-uniform triangulations if the components of $\boldsymbol{u}$ are in $H^2(\Omega)$. This raises the question under which circumstances one can guarantee this additional smoothness of the solution. The exact solution is for instance in $H^2(\Omega)$ if the domain is polygonal and convex and zero displacements are prescribed at the boundary [6]. However, for mixed boundary conditions and non-convex domains such statements cannot be made in general. This is especially true for structures containing cracks.

2.4. Extended finite element method

Often the approximation properties of the piecewise polynomial functions of the FEM are not sufficient to approximate the exact solution. Although the numerical solution converges for $h \to 0$ if the element angles are bounded from below, unreasonably small element sizes have to be used to obtain an acceptable solution. Examples for such problems are for instance structures containing re-entrant corners or cracks as discussed in the previous section. The stress and strain singularities cannot be resolved very well by piecewise polynomial ansatz functions. In the case of piecewise linear ansatz functions the stresses and strains of the numerical solution are constant inside each element. To reproduce a singularity with piecewise constant functions a very fine discretization must be used. Since the displacement components are not functions of H^2, the previous result about quadratic convergence does not hold any more.

But even if the displacement components are functions of H^2 the convergence can be slow. Imagine a smooth function which is a good approximation of a singular function. Although quadratic convergence is guaranteed

by theorem 2.5 and theorem 2.4, this convergence may only be experienced for extremely small values of h.

There are often situations in which it is possible to predict certain features of the exact solution solely based on the geometry of the structure and the boundary conditions. This a-priori knowledge about the solution can be used to improve the approximation properties of the ansatz space. Functions that are assumed to be a good approximation to the exact deformation of a structure can simply be added to the space if they are elements of $H^1_{\Gamma_D}$. It is obvious from equation (2.47) that adding new functions can only increase the approximation properties.

The partition of unity finite element method (PUFEM) introduced by Melenk and Babuška [7] is a special technique to enrich the FEM space using such a-priori knowledge about the solution. Let us say that a function ϕ which can be determined a-priori is assumed to have a good approximation property to the exact displacements of a structure in a certain area. Instead of adding ϕ directly to the ansatz space it is multiplied by a subset of the shape functions N_i. Each of the products is then used to enrich the function space. The shape functions N_i posses what is called the partition of unity property. That is, their sum is equal to 1 everywhere in Ω (to be fully precise, everywhere except inside those elements that are connected to Γ_D, but let us assume that shape functions N_i are defined for all nodes, in that case $\sum_i N_i = 1$ everywhere in Ω). Therefore, we have:

$$\sum_i N_i\phi = (\sum_i N_i)\phi = 1\phi = \phi \tag{2.57}$$

Hence, if ϕ is multiplied by all the shape functions the new space is able to reproduce the function ϕ.

Usually the approximation properties of ϕ are of a local character and only needed in a certain area of the structure. Therefore, if one uses those shape functions N_i whose support is close to the area of interest to calculate the products $N_i\phi$, ϕ can still be reproduced in that area.

The products $N_i\phi$ are called the nodal enrichments and ϕ is called the enrichment function. The corresponding nodes are called enriched nodes and the elements containing these nodes are called enriched elements. The nodal enrichments can be added as basis functions to the FEM ansatz space to

enhance the approximation properties.
There are several advantages to this approach if compared to the idea of adding ϕ directly to the FEM space. One advantage is that the support of the functions $N_i\phi$ is restricted and therefore the sparsity of the stiffness matrix is preserved. Another advantage is that the shape of ϕ can be adjusted locally.
Belytschko et al. [8] were the first ones to use this approach to reduce the burden of remeshing during the calculation of crack growth in elastic structures and named the new method X-FEM.
In crack growth calculations with the FEM, remeshing at each stage of crack growth is a necessary obstacle. Due to the singularities at the crack tip the mesh has to be refined several times depending on the location of the crack tip. Furthermore, element edges should align with the crack faces, otherwise the ansatz space is not able to reproduce the displacement jump along the crack.
But these are geometric restrictions on the mesh choice. If the crack geometry is complicated this may cause automatic mesh generators to break down.
In the X-FEM the structure can be meshed neglecting the presence of the crack. Therefore, the numerical solution of the FEM would not be able to represent the displacement jump along the crack. But in the X-FEM additional enrichment functions are used to compensate for this. A function ϕ that is discontinuous along the crack is defined. Each of the resulting products is used as an additional function in the FEM space. All nodes belonging to elements that are cut by the crack are enriched, except those of the element containing the crack tip.
A different type of enrichment is chosen to represent the displacement fields at the crack tip. As mentioned earlier the strains in the vicinity of the tip are singular. The problem of a two-dimensional crack in an infinite plane can be solved analytically. The analytical solution provides suitable enrichment functions for the area around the crack tip which are also called crack tip enrichments.
The partition of unity property of the shape functions can also be used to model cohesive cracks as demonstrated by Wells and Sluys [9].
In the first versions of X-FEM only the nodes of the element containing the

crack tip were enriched by the crack tip enrichments. Chahine et al. [10] however were able to show that optimal convergence rates could only be restored if all nodes inside an area around the crack tip which is independent of the element size are enriched. This is also called geometric enrichment.
Another class of problems for which remeshing is cumbersome are structures in which material interfaces are present. Generally the strains are discontinuous at the interface. The element edges have to align with the material interface, otherwise the strain jump cannot be reproduced exactly. If the geometry of the material interface is complicated, similar problems during mesh generation are experienced as in the presence of cracks. These problems can be circumvented by using an enrichment which has a discontinuous derivative [11]. As in the case of a crack, all nodes belonging to elements that are cut by the interface are enriched.
However, using such enrichments could not restore optimal convergence rates as it was achieved by the X-FEM in conjunction with cracks and geometric enrichments. The elements whose nodes are only partially enriched by a certain enrichment function are called blending elements. If the numerical solution is formed partially by the nodal enrichments, error terms are present in the blending elements. Fries [12] proposed to multiply the enrichment with a function called the ramp function. The ramp function is formed by adding the shape functions of all those nodes that usually would be enriched. Thus, the resulting ramp function takes values smaller or equal to one and greater or equal to zero. Fries then proposed to enrich all nodes whose shape functions have a support that coincides with the ramp function. Therefore, the number of enriched nodes increases if compared to standard X-FEM. This was called the corrected X-FEM. Optimal convergence rates could be achieved for structures containing material interfaces.
The corrected X-FEM can also be applied to the enrichments representing the crack tip combined with the geometric enrichment procedure. But the convergence rates cannot be improved any further. However, the error for a given mesh is significantly smaller.
We will use the methods mentioned above to obtain accurate numerical solutions for problems involving polycrystalline structures. This will simplify the meshing procedure significantly. As a consequence we are able to guarantee a successful mesh generation at each stage of crack growth. Therefore,

the crack growth simulations in section 4.4.2 can be automated completely. Please note that a successful mesh generation for complicated geometries can generally not be guaranteed if the mesh should meet the requirements mentioned above (i.e. alignment of the element edges with certain geometric features and refined meshes in certain areas). Furthermore, numerically determined enrichment functions will help us to represent strain singularities appropriately in the numerical solution.

X-FEM has been used in the context of polycrystalline structures and microstructural geometries before. Sukumar et al. [13] simulated crack growth in a polycrystalline structure using X-FEM. The same isotropic elastic material properties were assumed for each crystal, such that the standard crack tip enrichments for cracks in isotropic materials were appropriate enrichments. Moës et al. [14] used an enrichment with a discontinuous derivative to describe the non-smooth behavior of the solution at material interfaces in microstructural geometries. Simone et al. [15] described a method in which a Heaviside function was used to describe grain boundaries. Traction forces at the boundaries holding the grains together were added to the formulation.

Throughout this thesis plane strain conditions are assumed for the polycrystalline structures. Thus, the problem can be posed in two dimensions. A polycrystalline structure consists of different grains, where $\Omega_i \subset \mathbb{R}^2$ denotes the open domain covered by the i-th grain. Those domains do not overlap, but their closures cover Ω completely. Two neighboring grains are separated either by a grain boundary or by a crack. A crack may also be inside a grain. The situation is depicted in figure 2.3.

While a crack is a part of the boundary Γ_N, a grain boundary is part of the structural domain Ω. Points at which two or more grain boundaries terminate are called junctions. Cracks might also terminate there.

The constitutive relationship for the i-th grain is then given by:

$$\boldsymbol{\sigma} = \boldsymbol{D}_i : \boldsymbol{\epsilon}^{el} \tag{2.58}$$

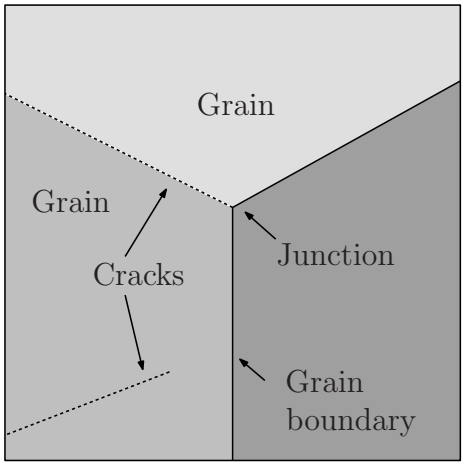


Figure 2.3.: *Example of a polycrystalline structure*

$\boldsymbol{D}_i$ is the fourth order elasticity tensor. The indicator function is defined by:

$$I_{\Omega_i}(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{x} \in \Omega_i \\ 0 & \text{otherwise} \end{cases} \tag{2.59}$$

The elasticity tensor $\boldsymbol{D}(\boldsymbol{x})$ for the whole structure can be defined by using indicator functions:

$$\boldsymbol{D}(\boldsymbol{x}) = \sum_i \boldsymbol{D}_i I_{\Omega_i}(\boldsymbol{x}) \tag{2.60}$$

In the following sections structural problems are not always interpreted as polycrystalline structures. However, the framework developed here makes it possible to treat those structures as special cases of a polycrystalline structure. For instance a structure formed by two isotropic materials can be interpreted as a polycrystalline structure formed by two grains with a special isotropic elasticity tensor.

To solve the problem numerically the structure is discretized by a set of finite elements. These elements do not need to conform with cracks or grain boundaries because appropriate enrichment functions are used. The X-FEM space used throughout this work is given by all displacement functions $\boldsymbol{u}^h$

with components of the form:

$$\boldsymbol{u}_k^h(\boldsymbol{x}) = \underbrace{\sum_{i=1}^{n_s} N_i(\boldsymbol{x}) a_{ik}}_{\text{FEM part}} + \underbrace{\sum_{j=1}^{n_e} \sum_{i \in e_j} N_i(\boldsymbol{x}) \phi_{ij}(\boldsymbol{x}) a_{ijk}^X}_{\text{X-FEM enrichment}} \tag{2.61}$$

$$k \in \{1, 2\} \tag{2.62}$$

$$e_j \subseteq \{1, ..., n_s\} \tag{2.63}$$

The a_{ik} and the a_{ijk}^X are unknown coefficients. The first term in equation (2.61) contains the standard finite element shape functions N_i. The second term contains the enrichment functions ϕ_{ij} multiplied by a subset of the polynomial shape functions. n_s is the number of polynomial shape functions and n_e is the number of enrichment functions. The set of nodal indices e_j is chosen individually for each enrichment.
By using different types of enrichment functions, the numerical solution is able to represent the following features of the exact solution appropriately:

- Displacement discontinuities along the crack;
- Strain discontinuities along the grain boundaries;
- Strain singularities at crack tips, junctions and re-entrant corners.

An enrichment scheme for each of the features mentioned above is now explained.

Crack interior enrichment

Let one of the cracks be denoted by Γ_C and the domains at the different sides of this crack by Ω_+ and Ω_-. We define the enrichment by using the Heaviside function [16]. If the j-th enrichment in equation (2.61) is chosen to represent the displacement jump along the crack interior, then it can be written as:

$$\phi_{ij}(\boldsymbol{x}) = H_{\Gamma_C}(\boldsymbol{x}) - H_{\Gamma_C}(\boldsymbol{x}_i) \tag{2.64}$$

$$H_{\Gamma_C}(\boldsymbol{x}) = \begin{cases} +1 \text{ if } \boldsymbol{x} \in \Omega_+ \\ -1 \text{ if } \boldsymbol{x} \in \Omega_- \end{cases} \tag{2.65}$$

$\boldsymbol{x}_i$ are the coordinates of the node associated with the i-th polynomial shape function. The subtraction of the value of $H(\boldsymbol{x}_i)$ is called shifting or nodal subtraction. This has several advantages. For example Dirichlet boundary conditions can be prescribed in the usual way, without being affected by the enrichments. Furthermore, the coefficients a_{ik} describe the displacements at the i-th node directly (without shifting, a summation of different coefficients would have to be performed to obtain this value). Shifting will be used for the other enrichments too.
Please note that there is no need to take special care of the blending elements for this type of enrichment, since the Heaviside function is piecewise constant. Therefore, an error introduced in the blending elements can be compensated by the polynomial shape functions.
Let the crack tip be denoted by $\boldsymbol{J}$. The set e_j is chosen depending on the support (supp$(\cdot)$) of the shape functions:

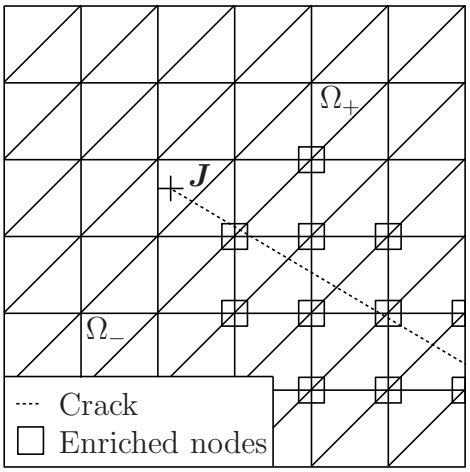
$$e_j = \{i : \text{supp}(N_i) \cap \Gamma_C \neq \emptyset \ \wedge \ \text{supp}(N_i) \cap \boldsymbol{J} = \emptyset\} \tag{2.66}$$

The situation is shown for a regular finite element mesh with a crack in figure 2.4(a). For branching cracks some adjustments have to be made. An approach different from the one used in this thesis was used by Daux et al. [17]. A special discontinuous junction function was introduced. The approach used in this thesis is slightly simpler to implement since only the definition of Heaviside functions is involved.
A situation in which three cracked grain boundaries join is shown in figure 2.5. Three different Heaviside functions can be defined by:

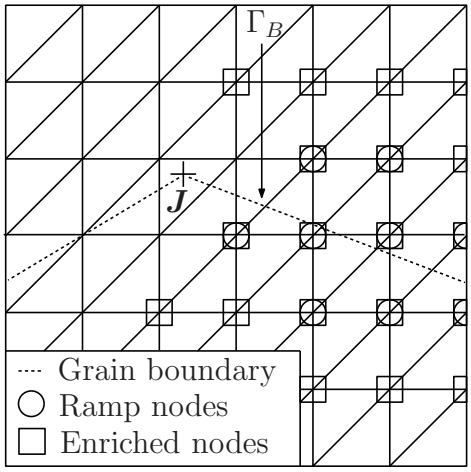
$$H_j = \begin{cases} +1 & \text{if } \boldsymbol{x} \in \Omega_j \\ -1 & \text{otherwise} \end{cases} \tag{2.67}$$

All nodes of the element are enriched ($e_j = \{1, 2, 3\}$). The displacement field for such an element is then:

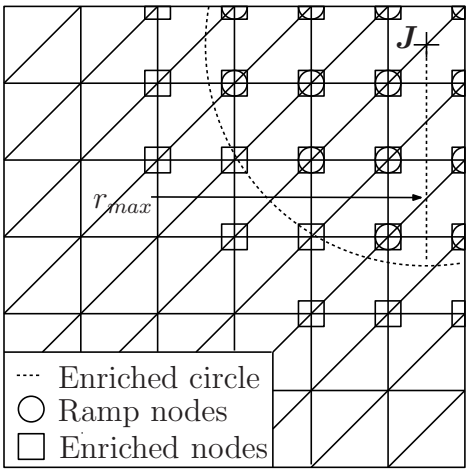
$$\begin{aligned} u_k(\boldsymbol{x}) \ &= \ \sum_{i=1}^{3} a_{ik} N_i(\boldsymbol{x}) + \\ &\quad \sum_{i \in e_j} \sum_j a^X_{ijk} N_i(\boldsymbol{x}) (H_j(\boldsymbol{x}) - H_j(\boldsymbol{x}_i)) \end{aligned} \tag{2.68}$$



(a) Enrichment scheme for the interior part of the crack



(b) Enrichment scheme for a grain boundary



(c) Enrichment scheme for a crack tip or a junction

Figure 2.4.: *Enriched nodes for different enrichment types*

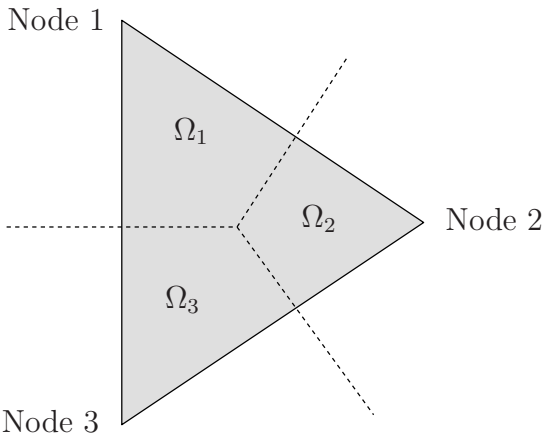


Figure 2.5.: *Three cracks joining at a common junction in an element*

If the kinematics of this element are represented correctly by the enrichment strategy, it should be possible to represent arbitrary displacement fields that are constant inside each domain Ω_j, but discontinuous along the cracks. In that case the domains Ω_j would be effectively disconnected.
By reordering the coefficients we can define:

$$\tilde{a}_{ik} = a_{ik} - \sum_j a^X_{ijk} H_j(\boldsymbol{x}_i) \tag{2.69}$$

Thus, the displacement field becomes:

$$u_k(\boldsymbol{x}) = \sum_{i=1}^{3} \tilde{a}_{ik} N_i(\boldsymbol{x}) + \sum_{i \in e_j} \sum_j a^X_{ijk} N_i(\boldsymbol{x}) H_j(\boldsymbol{x}) \tag{2.70}$$

The discussion is simplified by assuming:

$$\tilde{a}_{1k} = \tilde{a}_{2k} = \tilde{a}_{3k} \tag{2.71}$$

$$a^X_{1jk} = a^X_{2jk} = a^X_{3jk} \quad \forall j \tag{2.72}$$

Since the shape functions form a partition of unity, the displacement field inside the element becomes:

$$u_k(\boldsymbol{x}) \;\; = \;\; \tilde{a}_{1k} + \sum_j a^X_{1jk} H_j(\boldsymbol{x}) \tag{2.73}$$

Furthermore, let us assume that an arbitrary piecewise constant displacement field is given by constants c_{jk}, where c_{jk} is the value of the k-th component inside the domain Ω_j. Due to the definition of H_j, one has to solve the following equation system to represent this field accurately by (2.73):

$$\begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} \tilde{a}_{1k} \\ a^X_{11k} \\ a^X_{12k} \\ a^X_{13k} \end{bmatrix} = \begin{bmatrix} c_{1k} \\ c_{2k} \\ c_{3k} \end{bmatrix} \tag{2.74}$$

Obviously there are infinitely many solutions. Neglecting one of the Heaviside functions H_j in the enrichment procedure would result in an equation system with one less column, which would be non-singular for any choice of H_j. Hence, we may delete any of the last three columns. From this we can conclude that the kinematics of an element containing a cracked triple junction are represented correctly if we use two of the three Heaviside functions defined above. Please note that this result is independent of the angles between the cracks and the position of the junction. Similar conclusions can be made for an element in which four or two cracks terminate at a common junction.

Grain boundary enrichment

Let one of the grain boundaries be denoted by Γ_B and the junction to which this grain boundary is connected by $\boldsymbol{J}$. The solution will in general have a discontinuous derivative along the boundary. Therefore, we use a function for the construction of the enrichment, which has the same property. The distance function [11] for the grain boundary Γ_B can be defined by:

$$D_{\Gamma_B}(\boldsymbol{x}) = \min_{\hat{\boldsymbol{x}} \in \Gamma_B} \|\hat{\boldsymbol{x}} - \boldsymbol{x}\| \tag{2.75}$$

To avoid problems in blending elements the distance function is multiplied by a ramp function [12]. If the j-th enrichment in equation (2.61) is chosen

to represent this grain boundary, then it can be written as:

$$\phi_{ij}(\boldsymbol{x}) = \left[D_{\Gamma_B}(\boldsymbol{x}) - D_{\Gamma_B}(\boldsymbol{x}_i)\right] R(\boldsymbol{x}) \tag{2.76}$$

$$R(\boldsymbol{x}) = \sum_{i \in r_j} N_i(\boldsymbol{x}) \tag{2.77}$$

$$r_j = \{i : \text{supp}(N_i) \cap \Gamma_B \neq \emptyset \wedge \ \text{supp}(N_i) \cap (\boldsymbol{J}) = \emptyset\} \tag{2.78}$$

The nodes associated with the set r_j are called ramp nodes. The set e_j for the grain boundary enrichment is given by:

$$e_j = \{i : \text{supp}(N_i) \cap \text{supp}(R) \neq \emptyset \ \wedge \ \text{supp}(N_i) \cap (\boldsymbol{J}) = \emptyset\} \tag{2.79}$$

The situation is depicted in figure 2.4(b).

Enrichment at crack tips, junctions and re-entrant corners

In the case of a junction, a crack tip or a re-entrant corner the enrichment is determined numerically. The procedure is described in section 3.2. The enrichment is written in polar coordinates with radius r and angle θ, where the pole of the coordinate system is located at the junction, the corner or the crack tip respectively. The enrichments considered here can be decomposed in a term depending on the radius and a term depending on the angle:

$$P(\boldsymbol{x}) = \Re\left[r(\boldsymbol{x})^{\lambda} \psi(\theta(\boldsymbol{x})\right] \quad \lambda \in \mathbb{C} \tag{2.80}$$

The enrichments are again generated by shifting and using a ramp function. If the j-th enrichment in equation (2.61) is chosen to represent a junction, a re-entrant corner or a crack tip at coordinates $\boldsymbol{J}$, then it can be written as:

$$\phi_{ij}(\boldsymbol{x}) = \left[P(\boldsymbol{x}) - P(\boldsymbol{x}_i)\right] R(\boldsymbol{x}) \tag{2.81}$$

$$R(\boldsymbol{x}) = \sum_{i \in r_j} N_i(\boldsymbol{x}) \tag{2.82}$$

$$r_j = \{i : \|\boldsymbol{x}_i - \boldsymbol{J}\| \leq r_{max}\} \tag{2.83}$$

$$e_j = \{i : \text{supp}(N_i) \cap \text{supp}(R) \neq \emptyset \ \} \tag{2.84}$$

r_{max} is a radius independent of the mesh size, therefore, the enriched area does not decrease during mesh refinement. Since this is necessary to obtain optimal convergence in the case of a crack [10], it is an important requirement for the more general case discussed here. The situation is shown in figure 2.4(c).

2.5. Iterative solvers

If the equation systems become large a direct solution is not efficient any more. The computational effort involved in solving equation systems directly via matrix decomposition grows cubically. This is also true for sparse matrices, since generally the sparsity pattern is not preserved during a matrix decomposition. Iterative solvers improve the solution stepwise. The solver stops if the residual becomes sufficiently small.
Different iterative solvers are available. One of them is the minimal residual method (MINRES). This method can be applied to symmetric indefinite matrices. Generally the stiffness matrix in the FEM and the X-FEM is symmetric positive definite, which would allow to use other solvers like the conjugate gradient method (CG) too. But the preconditioning technique in section 3.3 results in symmetric indefinite matrices. This is due to the addition of constraints in the equation systems. Therefore the MINRES solver was chosen for the examples considered in this thesis.
We will start the discussion with the generalized minimal residual method (GMRES). Simplifications of this method can be made if the matrix is symmetric. These simplifications will lead us to the MINRES solver.
We want to solve:

$$\boldsymbol{K}\boldsymbol{a} = \boldsymbol{f} \tag{2.85}$$

The task can be interpreted as minimizing the norm of the residual in $\mathbb{R}^n$:

$$\min_{\boldsymbol{a}\in\mathbb{R}^n} \|\boldsymbol{K}\boldsymbol{a} - \boldsymbol{f}\| \tag{2.86}$$

In the i-th step of the GMRES solver the residual is minimized over an i-dimensional subspace of $\mathbb{R}^n$. Assume that the columns of the matrix

$\boldsymbol{Q}_i \in \mathbb{R}^{n \times i}$ form an orthogonal basis of this space:

$$\boldsymbol{Q}_i = \Big[\boldsymbol{q}_1, ..., \boldsymbol{q}_i\Big] \tag{2.87}$$

Then the problem becomes:

$$\min_{\hat{\boldsymbol{a}}_i \in \mathbb{R}^i} \|\boldsymbol{K}\boldsymbol{Q}_i\hat{\boldsymbol{a}}_i - \boldsymbol{f}\| \tag{2.88}$$

A solution can be obtained by solving:

$$\boldsymbol{Q}_i^T\boldsymbol{K}\boldsymbol{Q}_i\hat{\boldsymbol{a}}_i = \boldsymbol{Q}_i^T\boldsymbol{f} \tag{2.89}$$

The space that the GMRES method uses is called the Krylov space. For its definition a vector and a matrix is needed:

Definition 2.1 *The i-th Krylov space for a matrix* $\boldsymbol{K}$ *and a vector* $\boldsymbol{f}$ *is defined by:*

$$\mathcal{K}(\boldsymbol{K}, \boldsymbol{f})^i = span\{\boldsymbol{f}, \boldsymbol{K}\boldsymbol{f}, ..., \boldsymbol{K}^{i-1}\boldsymbol{f}\} \tag{2.90}$$

In the i-th step the equations of the GMRES solver can be written as:

$$\boldsymbol{K}\boldsymbol{Q}_i = [\boldsymbol{Q}_i, \boldsymbol{q}_{i+1}]\,\boldsymbol{T}_i \tag{2.91}$$

$\boldsymbol{q}_{i+1}$ is a column vector and the matrix $\boldsymbol{T}_i$ is an element of $\mathbb{R}^{i+1,i}$. The vector $\boldsymbol{q}_{i+1}$ is orthogonal to the columns of $\boldsymbol{Q}_i$. To obtain the equations in the $i+1$-th step we set:

$$\boldsymbol{Q}_{i+1} := [\boldsymbol{Q}_i, \boldsymbol{q}_{i+1}] \tag{2.92}$$

We obtain $\boldsymbol{q}_{i+2}$ by:

$$\hat{\boldsymbol{q}}_{i+2} := \boldsymbol{K}\boldsymbol{q}_{i+1} - \sum_{j=1}^{i+1} \left(\boldsymbol{q}_j^T\boldsymbol{K}\boldsymbol{q}_{i+1}\right)\boldsymbol{q}_j \tag{2.93}$$

$$\boldsymbol{q}_{i+2} := \frac{\hat{\boldsymbol{q}}_{i+2}}{\|\hat{\boldsymbol{q}}_{i+2}\|} \tag{2.94}$$

Please note that the matrix-vector product $\boldsymbol{K}\boldsymbol{q}_{i+1}$ in (2.93) should be calculated only once.

We define:

$$\boldsymbol{t}_{i+1} := \begin{bmatrix} \boldsymbol{q}_1^T \boldsymbol{K} \boldsymbol{q}_{i+1} \\ \vdots \\ \boldsymbol{q}_{i+1}^T \boldsymbol{K} \boldsymbol{q}_{i+1} \end{bmatrix} \tag{2.95}$$

Then the matrix $\boldsymbol{T}_{i+1}$ is obtained by:

$$\boldsymbol{T}_{i+1} := \begin{bmatrix} \boldsymbol{T}_i & \boldsymbol{t}_{i+1} \\ \boldsymbol{0} & \|\hat{\boldsymbol{q}}_{i+2}\| \end{bmatrix} \tag{2.96}$$

Therefore, all the matrices and vectors in (2.91) are defined for the step $i+1$ by using quantities from step i.
Assume we start the process with:

$$\boldsymbol{q}_1 := \frac{\boldsymbol{f}}{\|\boldsymbol{f}\|} \tag{2.97}$$

Then $\boldsymbol{Q}_i$ is a basis of the Krylov-space $\mathcal{K}(\boldsymbol{K}, \boldsymbol{f})^i$ for every i (except for the case $\hat{\boldsymbol{q}}_i = \boldsymbol{0}$ for some i, which is known as a break down). This can be shown by induction: If $\boldsymbol{Q}_{i+1}$ is a basis of $\mathcal{K}(\boldsymbol{K}, \boldsymbol{f})^{i+1}$ then $\boldsymbol{q}_{i+2}$ is an element of $\mathcal{K}(\boldsymbol{K}, \boldsymbol{f})^{i+2}$ which follows from the definition of the Krylov-space and equations (2.93) and (2.94).
$\boldsymbol{Q}_i$ for every i forms an orthogonal basis of $\mathcal{K}(\boldsymbol{K}, \boldsymbol{f})^i$ because equation (2.93) basically describes a Gram-Schmidt procedure applied to the new basis vectors of the Krylov space.
Multiplication of equation (2.91) with $\boldsymbol{Q}_i^T$ from the left results in:

$$\boldsymbol{Q}_i^T \boldsymbol{K} \boldsymbol{Q}_i = \hat{\boldsymbol{T}}_i \tag{2.98}$$

$\hat{\boldsymbol{T}}_i$ contains the first i rows of $\boldsymbol{T}_i$. The approximate solution $\boldsymbol{a}_i$ in (2.89) can therefore be obtained by solving:

$$\hat{\boldsymbol{T}}_i \hat{\boldsymbol{a}}_i = \boldsymbol{Q}_i^T \boldsymbol{f} \tag{2.99}$$

By the definition of the algorithm the matrix $\hat{\boldsymbol{T}}_i$ looks like:

$$\hat{\boldsymbol{T}}_i = \begin{bmatrix} * & \dots & \dots & \dots & * \\ * & & & & \vdots \\ 0 & \ddots & & & \vdots \\ \vdots & \ddots & \ddots & & \vdots \\ 0 & \dots & 0 & * & * \end{bmatrix} \tag{2.100}$$

Matrices of this structure are also known as Hessenberg matrices. Looking at equation (2.98) we see that symmetry of $\boldsymbol{K}$ would imply symmetry of $\hat{\boldsymbol{T}}_i$ for every i. Therefore, the Hessenberg matrix becomes a tridiagonal matrix for symmetric $\boldsymbol{K}$. But in that case the number of inner products that have to be calculated in (2.93) can be reduced significantly since most of them are zero by default. Leaving out those operations in the GMRES procedure would effectively be the MINRES solver for symmetric indefinite matrices in its simplest form.
The residual in the i-th step is defined by:

$$\boldsymbol{r}_i := \|\boldsymbol{K}\boldsymbol{Q}_i\hat{\boldsymbol{a}}_i - \boldsymbol{f}\| \tag{2.101}$$

The relationship of the residual to the spectrum of the matrix $\boldsymbol{K}$ is stated in the next theorem:

Theorem 2.6 *Let $\boldsymbol{Q}_{\boldsymbol{K}}$ be an orthogonal set of eigenvectors of the symmetric matrix $\boldsymbol{K}$. Furthermore, let $\lambda_1, ..., \lambda_n$ be the eigenvalues of $\boldsymbol{K}$. If P^i is the space of polynomials of order i, then the norm of the residual $\boldsymbol{r}_i$ in the i-th step of the MINRES-solver is bounded by:*

$$\|\boldsymbol{r}_i\| \leq \min_{p \in P^i \mid p(0)=1} \max_j p(\lambda_j) \|\boldsymbol{Q}_{\boldsymbol{K}}^T \boldsymbol{f}\| \tag{2.102}$$

Proof Due to the definition of the Krylov space we can write the residual in the i-th step of the MINRES solver as:

$$\|\boldsymbol{K}(p_1\boldsymbol{f} + p_2\boldsymbol{K}\boldsymbol{f} + p_3\boldsymbol{K}^2\boldsymbol{f} + ...) - \boldsymbol{f}\| \tag{2.103}$$

$p_1, ..., p_i$ are some coefficients. If $\boldsymbol{Q}_{\boldsymbol{K}}$ is a set of orthogonal eigenvectors of

$\boldsymbol{K}$ and $\boldsymbol{\Lambda}$ is with diagonal matrix with eigenvalues $\lambda_1, ..., \lambda_n$ on its diagonal, the residual can be written as:

$$\|\boldsymbol{Q_K}\boldsymbol{\Lambda}\boldsymbol{Q_K}^T(p_1\boldsymbol{f} + p_2\boldsymbol{Q_K}\boldsymbol{\Lambda}\boldsymbol{Q_K}^T\boldsymbol{f} + p_3(\boldsymbol{Q_K}\boldsymbol{\Lambda}\boldsymbol{Q_K}^T)^2\boldsymbol{f} + ...) - \boldsymbol{f}\| \tag{2.104}$$

$$=\|\boldsymbol{Q_K}\boldsymbol{\Lambda}\boldsymbol{Q_K}^T(p_1\boldsymbol{f} + p_2\boldsymbol{Q_K}\boldsymbol{\Lambda}\boldsymbol{Q_K}^T\boldsymbol{f} + p_3\boldsymbol{Q_K}\boldsymbol{\Lambda}^2\boldsymbol{Q_K}^T\boldsymbol{f} + ...) - \boldsymbol{f}\| \tag{2.105}$$

$$=\|\boldsymbol{\Lambda}\boldsymbol{Q_K}^T(p_1\boldsymbol{f} + p_2\boldsymbol{Q_K}\boldsymbol{\Lambda}\boldsymbol{Q_K}^T\boldsymbol{f} + p_3\boldsymbol{Q_K}\boldsymbol{\Lambda}^2\boldsymbol{Q_K}^T\boldsymbol{f} + ...) - \boldsymbol{Q_K}^T\boldsymbol{f}\| \tag{2.106}$$

$$=\|p_1\boldsymbol{\Lambda}\boldsymbol{Q_K}^T\boldsymbol{f} + p_2\boldsymbol{\Lambda}^2\boldsymbol{Q_K}^T\boldsymbol{f} + p_3\boldsymbol{\Lambda}^3\boldsymbol{Q_K}^T\boldsymbol{f} + ... - \boldsymbol{Q_K}^T\boldsymbol{f}\| \tag{2.107}$$

$$=\|(p_1\boldsymbol{\Lambda} + p_2\boldsymbol{\Lambda}^2 + p_3\boldsymbol{\Lambda}^3 + ... - \boldsymbol{I})\boldsymbol{Q_K}^T\boldsymbol{f}\| \tag{2.108}$$

$$=\|(\boldsymbol{I} - p_1\boldsymbol{\Lambda} - p_2\boldsymbol{\Lambda}^2 - p_3\boldsymbol{\Lambda}^3 - ...)\boldsymbol{Q_K}^T\boldsymbol{f}\| \tag{2.109}$$

$$\leq \max_j p(\lambda_j)\|\boldsymbol{Q_K}^T\boldsymbol{f}\| \tag{2.110}$$

We set $p_0 := 1$ to obtain the polynomial p. This proves the theorem since the MINRES iterates minimize the residual norm over the i-th Krylov space. □

According to this theorem, the distribution of eigenvalues on the real axis plays an important role if the convergence of the MINRES solver is discussed. If we consider symmetric positive definite matrices only, we can interpret the theorem as follows: The MINRES convergence will be fast if the interval on the real axis which contains all eigenvalues is small and its distance to zero is large. It is easy to imagine a parabola with function value 1 at $x = 0$ and function value 0 somewhere inside that interval. The polynomial p in (2.102) would take small values for all λ_i, thus, the expression on the right would be small. Therefore, in the second step of the solution process a good approximation would be obtained already. Similar statements can be made for indefinite matrices.
We may conclude that the eigenvalue with the smallest magnitude and the one with the largest magnitude are more important than the other eigenvalues since they determine the interval mentioned above. The eigenvalue with the smallest magnitude should have a large magnitude, while the eigenvalue with the largest magnitude should be of a small magnitude. Both properties of the spectrum are reflected in the condition number:

Definition 2.2 *Let λ_{max} and λ_{min} be the eigenvalues with the largest and the smallest magnitude of a symmetric matrix $\boldsymbol{K}$. The condition number*

$\kappa(\boldsymbol{K})$ of $\boldsymbol{K}$ is defined by:

$$\kappa(\boldsymbol{K}) := \frac{|\lambda_{max}|}{|\lambda_{min}|} \tag{2.111}$$

The following theorem relates the condition number to the convergence behavior of the MINRES solver:

Theorem 2.7 *The norm of the residual $\boldsymbol{r}_i$ in the i-th step of the MINRES solver for a symmetric matrix $\boldsymbol{K}$ is bounded by*

$$\|\boldsymbol{r}_i\| \leq \left[1 - \kappa(\boldsymbol{K})^{-2}\right]^{\frac{i}{2}} \|\boldsymbol{r}_0\| \tag{2.112}$$

Proof In a more general form the theorem was proven by Eisenstat et al. [18]. □

Therefore, if the condition number is big, the expression in brackets is only slightly smaller than 1 and the convergence is slow.

3. X-FEM for polycrystalline structures

In section 2.4 we collected several concepts and ideas from the X-FEM literature which were put together to form a general framework for solving mechanical problems involving cracked polycrystalline structures numerically. This framework, however, is not yet capable of performing the lifetime prediction procedure in chapter 4 in an efficient manner. In this chapter several methods are proposed and tested to further develop it.

The core of this chapter is formed by section 3.2, section 3.3 and section 3.4. In section 3.2 a methodology to determine weakly singular enrichment functions for arbitrary two-dimensional structures will be presented. In section 3.3 a preconditioning technique is developed which enables a fast solution of X-FEM equation systems for arbitrary enrichment strategies. In section 3.4 a meshing strategy is described and tested which, in combination with the X-FEM, makes a complete automation of the crack growth calculations in chapter 4 possible.

We start this chapter with a discussion of the numerical integration procedure employed throughout this thesis. Although the influence of the numerical integration scheme is not directly visible in the numerical experiments performed in this chapter, this topic has to be discussed first, since all the other concepts are built on it.

Towards the end of this chapter an implementation of the penalty method for X-FEM which prevents crack faces from overlapping is explained. This is done in section 3.5, the main challenge is to combine the penalty method with the preconditioning technique developed earlier.

Finally, the combination of the X-FEM procedure explained so far with

an explicit solver for transient problems involving creep deformation is discussed.

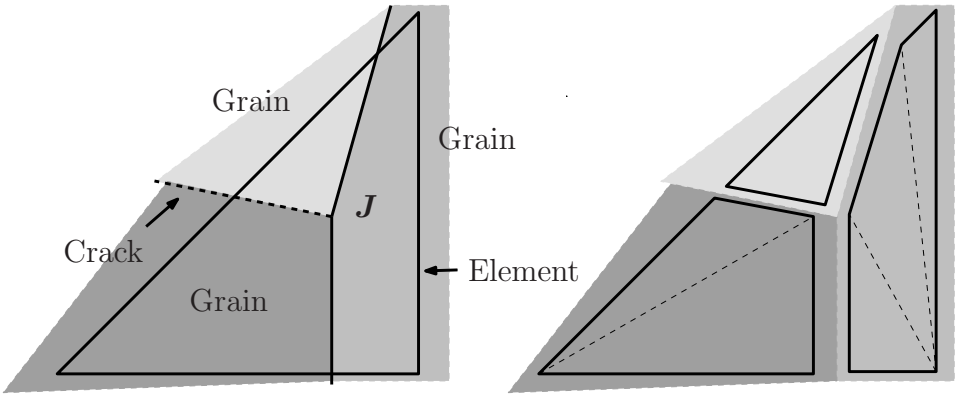
3.1. Numerical integration

The necessity to modify the standard Gauss integration employed for finite elements for the X-FEM was realized early. Moës et al. [16] decomposed the finite elements into smaller triangles and performed Gauss integration of a higher order for each of them. A similar approach will be used here.
More advanced integration procedures were proposed by Béchet et al. [10]. An integral transformation was used to integrate weakly singular functions. The number of function evaluations in the integration procedure could be reduced significantly.
Other approaches were discussed by Natarajan et al. [19]. One of these approaches transforms the integral onto the boundary of the finite element. This eliminates the need for subdividing the elements into integration domains.
Numerical integration of a function f over some area A starts with placing a number of integration points $\boldsymbol{g}_i$ inside the area. A weight w_i is assigned to each point. An approximation to the exact integral is then:

$$\int_A f da \approx \sum_i w_i f(\boldsymbol{g}_i) \tag{3.1}$$

If A is a triangle the w_i and the $\boldsymbol{g}_i$ can be chosen such that (3.1) holds exactly for polynomials of a certain order. In Gaussian quadrature the integration points and weights are chosen such that the polynomial order is maximal.
The entries of the element stiffness matrices of standard finite elements can be integrated exactly (at least if round-off errors are neglected). For triangular elements and piecewise linear shape functions one integration point is sufficient since the integration has to be performed over a constant function. Thus, the weight is simply the area of the triangle. Generally a Gaussian quadrature of order n integrates polynomials up to an order of $2n-1$ exactly. In one dimension the order of the quadrature is equal to the number of integration points. In two dimensions the number of integration points

is generally larger than the order (order one is the exception). A good introduction to Gaussian quadrature was given by Schwarz and Köckler [20]. The enrichments however may not be polynomials. Therefore, numerical integration by Gaussian quadrature will only be an approximation. But we may hope for a small integration error if we increase the order of integration. The error is dependent on the smoothness of the function that is to be integrated [21]. The enrichments will be non-smooth across the grain boundaries and junctions. Thus, the accuracy of the numerical integration will be higher if the elements are split in integration domains which do not contain junctions, boundaries or cracks in their interior.
In the applications in section 4.4.2 the structures are formed by grains and cracks will only appear at the grain boundaries. The element domain is decomposed into integration subdomains formed by the intersection with the neighboring grains. As a result we obtain polygonal domains which do



(a) Element cut by grain boundaries and a crack

(b) Decomposition of the element domain into subtriangles

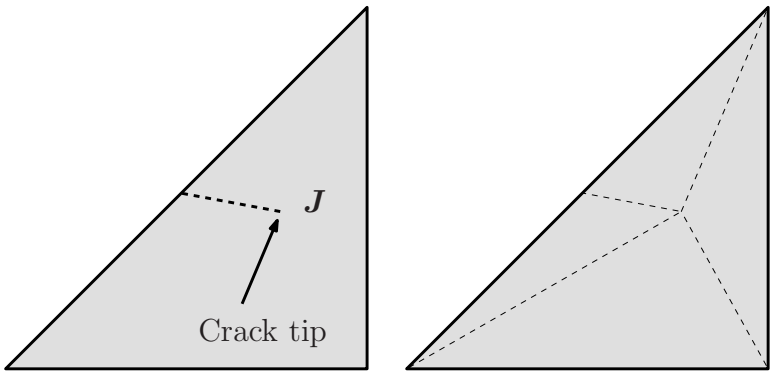
Figure 3.1.: *Decomposition of the element domain into integration domains*

not contain cracks or boundaries in their interior. The situation is shown in figure 3.1(a).
The polygonal domains are then decomposed into subtriangles as shown in figure 3.1(b). Inside these subtriangles we can assume that the enrichments are sufficiently smooth.
Gaussian integration is performed for each subtriangle. The integration order for the subtriangles is shown in table 3.1. If the enrichment functions stem from a junction enrichment and the junction is in the interior of the

Subtriangles in an element containing an enriched junction	15
Subtriangles in an enriched element	10

Table 3.1.: *Integration order for the subtriangles in the enriched elements*

corresponding element, the integration order is increased.
In the applications in section 4.4.2 some elements are not fully part of the structural domain. In that case the integration is only performed over the interior part.
In some of the following sections (straight) cracks appear not only at grain boundaries, but also in the interior of a grain. In that case an element cut by the crack is either cut completely, or it contains the crack tip. If it is cut completely, the elements domain can be decomposed into two polygonal domains which are then further decomposed into subtriangles as already discussed. If the element contains a crack tip, the elements domain is split



(a) Element with a crack tip in its interior

(b) Decomposition in integration domains

Figure 3.2.: *Decomposition of the element domain into integration domains if a crack is in the grain interior*

as shown in figure 3.2. The crack forms a straight line inside the element domain. Other straight lines can be drawn from the crack tip to each of the elements nodes. All of these lines are used to decompose the element into triangular integration domains.
This integration procedure works well if grain boundaries and cracks are straight lines. For polycrystalline structures obtained by a Voronoi-tessellation

as in chapter 4 this is true.

3.2. Numerically determined enrichment functions

In this section it is explained how enrichment functions can be used to enhance the approximation properties of the FEM if weak singularities are present in the exact solution. Weak singularities may arise at crack tips, junctions or re-entrant corners. The standard enrichment functions at the crack tip used by Belytschko et al. [22] are examples of such enrichments. They can be determined analytically, but they are only a good representation of the asymptotic fields in the vicinity of the crack tip if the material is isotropic. In the case of general strain singularities it is not always possible to determine suitable enrichment functions analytically. The asymptotic fields are dependent on the surrounding materials and the geometry of the structure.
In this section the asymptotic fields are determined numerically and used in equation (2.81). This idea was first published by Menk and Bordas [23]. Numerical experiments are performed to show that these enrichments are able to restore optimal convergence rates if weak singularities are present in the exact solution. This will complete the enrichment strategy described in section 2.4.
Other enrichment strategies for situations in which the explicit form of the asymptotic fields is not known analytically exist in the literature. The Spider-X-FEM developed by Chahine et al. [24] is one of them. However, the order of the singularity has to be known in advance.
A method to use the idea of enrichment if only very little a-priori knowledge is available was proposed by Waismann and Belytschko [25]. The idea is to enrich with functions called parametric enrichments, which depend on unknown parameters. Those parameters are adjusted during the course of the algorithm by minimizing an a-posteriori error estimate from a previous calculation. This idea could also be used to model weak singularities in polycrystalline structures by enrichments, but the equation systems have to be solved several times to obtain a solution which minimizes the error.

3.2.1. Determination of the enrichment functions

General linear elastic problem description at crack tips, junctions and re-entrant corners

Li et al. [26] describe a way to determine the singular stress distribution in the vicinity of a notch formed by different anisotropic materials. The situation is depicted in figure 3.3. The equations (3.2)-(3.17) are reproduced

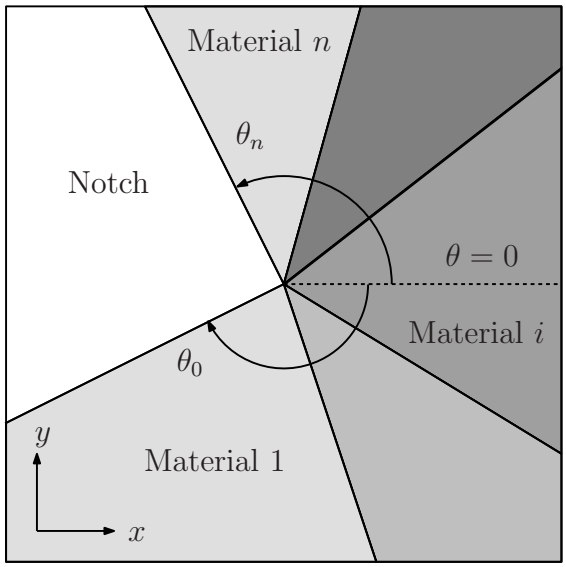


Figure 3.3.: *Notch formed by different anisotropic material wedges*

here from the original work for two reasons. The first one is to motivate the choice for a certain discretization parameter m used in the numerical experiments. The second reason is to explain how this approach can be used if the angle of the notch in figure 3.3 vanishes and all interfaces are perfectly bonded, which was not discussed in the original paper.
The displacements and the stresses are written in cylindrical coordinates, the origin being located at the tip of the notch. The equations governing the plane strain problem can then be written in terms of the variable vector $\boldsymbol{\zeta}$:

$$\boldsymbol{\zeta} := [\boldsymbol{\zeta}_1 \ \boldsymbol{\zeta}_2]^T \tag{3.2}$$

$$\boldsymbol{\zeta}_1 := [u_\theta \ u_r \ u_z] \tag{3.3}$$

$$\boldsymbol{\zeta}_2 := [\sigma_\theta r \ \tau_{r\theta} r \ \tau_{\theta z} r] \tag{3.4}$$

r, θ and z are the radial, the angular and the out-of-plane component of the cylindrical coordinate system. u_θ, u_r and u_z are the corresponding displacement components in cylindrical coordinates. Transforming the stress tensor from euclidean to cylindrical coordinates gives the normal stress for the radial component σ_θ and the shear stresses $\tau_{r\theta}$ and $\tau_{\theta z}$.
The solution is assumed to be of the form:

$$\boldsymbol{\zeta}(r,\theta) = r^\lambda \boldsymbol{\psi}(\theta) \tag{3.5}$$

The exponent λ and the function $\boldsymbol{\psi}(\theta)$, which we will call angular function, are unknown. λ must be greater than zero, otherwise the elastic energy would be infinite.
Using this assumption the equilibrium equations depend on θ and λ in the following way:

$$\frac{\partial \boldsymbol{\psi}(\theta)}{\partial \theta} = \boldsymbol{H}(\theta,\lambda)\boldsymbol{\psi}(\theta) \tag{3.6}$$

$\boldsymbol{H}(\theta,\lambda)$ is a matrix whose entries depend nonlinearly on θ and λ. The derivation of $\boldsymbol{H}(\theta,\lambda)$ is rather technical and can be found in Appendix A.
In the case of a polycrystalline structure, the different materials in figure 3.3 would be grains. A crack terminating at a junction would be described in this framework simply by a notch with a zero angle.
The entries of the matrix $\boldsymbol{H}(\theta,\lambda)$ are different in the plane stress and the plane strain case, however, we are only interested in the plane strain case.

Determination of the order of the singularity

The order of the singularity λ is obtained as follows. Suppose that for the i-th material the interfaces to the neighboring materials are located at θ_i and θ_{i+1}. The vectors $\boldsymbol{\psi}(\theta_i)$ and $\boldsymbol{\psi}(\theta_{i+1})$ can be related by using equation (3.6). Assume the interval $[\theta_i, \theta_{i+1}]$ is divided into m smaller intervals and let the j-th of these intervals be given by $[\theta_i^j, \theta_i^{j+1}]$ and $h = \theta_i^{j+1} - \theta_i^j$ be the interval angle. By using the trapezoidal rule the following approximation can be made for small intervals:

$$\boldsymbol{\psi}(\theta_i^{j+1}) - \boldsymbol{\psi}(\theta_i^j) \approx \left[\boldsymbol{H}(\theta_i^j,\lambda)\boldsymbol{\psi}(\theta_i^j) + \boldsymbol{H}(\theta_i^{j+1},\lambda)\boldsymbol{\psi}(\theta_i^{j+1})\right]\frac{h}{2} \tag{3.7}$$

Solving this equation for $\boldsymbol{\psi}(\theta_i^{j+1})$ yields:

$$\boldsymbol{\psi}(\theta_i^{j+1}) = \boldsymbol{B}_i^j \boldsymbol{\psi}(\theta_i^j) \tag{3.8}$$

$$\boldsymbol{B}_i^j = \left[\boldsymbol{I} - \boldsymbol{H}(\theta_i^{j+1}, \lambda)\frac{h}{2}\right]^{-1} \left[\boldsymbol{I} + \boldsymbol{H}(\theta_i^j, \lambda)\frac{h}{2}\right] \tag{3.9}$$

Thus, one obtains for $\boldsymbol{\psi}(\theta_{i+1})$ and $\boldsymbol{\psi}(\theta_i)$:

$$\boldsymbol{\psi}(\theta_{i+1}) = \boldsymbol{B}_i \boldsymbol{\psi}(\theta_i) \tag{3.10}$$

$$\boldsymbol{B}_i = \prod_{j=1}^{m} \boldsymbol{B}_i^{m-j} \tag{3.11}$$

The vector $\boldsymbol{\zeta}$ is continuous across interfaces [26]. Thus, the vectors $\boldsymbol{\psi}(\theta_0)$ and $\boldsymbol{\psi}(\theta_n)$ can be related by:

$$\boldsymbol{\psi}(\theta_n) = \boldsymbol{B}(\lambda)\boldsymbol{\psi}(\theta_0) \tag{3.12}$$

$$\boldsymbol{B}(\lambda) = \prod_{i=1}^{n} \boldsymbol{B}_{n-i} \tag{3.13}$$

By using the definition of $\boldsymbol{\zeta}$ we can write:

$$\begin{bmatrix} \boldsymbol{\zeta}_1(\theta_n) \\ \boldsymbol{\zeta}_2(\theta_n) \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}_{11}(\lambda) & \boldsymbol{B}_{12}(\lambda) \\ \boldsymbol{B}_{21}(\lambda) & \boldsymbol{B}_{22}(\lambda) \end{bmatrix} \begin{bmatrix} \boldsymbol{\zeta}_1(\theta_0) \\ \boldsymbol{\zeta}_2(\theta_0) \end{bmatrix} \tag{3.14}$$

The submatrices $\boldsymbol{B}_{11}$,$\boldsymbol{B}_{12}$,$\boldsymbol{B}_{21}$ and $\boldsymbol{B}_{22}$ are complex 3×3-matrices.
Choosing zero-traction boundary conditions at the free edges yields:

$$\boldsymbol{\zeta}_2(\theta_0) = \boldsymbol{p}(\theta_n) = 0 \tag{3.15}$$

Of course the equation (3.14) is fulfilled if $\boldsymbol{\zeta}_1(\theta_0) = \boldsymbol{\zeta}_1(\theta_n) = \boldsymbol{0}$. But to obtain non-trivial solutions fulfilling the boundary conditions, we are looking for a non-zero vector that combines the columns of $\boldsymbol{B}_{21}$ to zero. Thus, a nonlinear eigenvalue-problem in λ has to be solved:

$$\det(\boldsymbol{B}_{21}(\lambda)) = 0 \tag{3.16}$$

The solutions with $0 < \Re[\lambda] < 1$ are the ones describing a singular strain state, which is obvious from differentiating equation (3.5) with respect to r. If an eigenpair $(\lambda, \boldsymbol{\zeta}_1)$ is found, then by using equation (3.15) the vector $\boldsymbol{\psi}(\theta_n)$ is:

$$\boldsymbol{\psi}(\theta_n) = \begin{bmatrix} \boldsymbol{\zeta}_1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.17}$$

To solve the eigenvalue problem a Newton-type method has to be used because of the non-linearity. The difficulty is to choose appropriate start values. To describe a singular behavior in the structure correctly, all eigenvalues have to be found. Complex eigenvalues might appear. But since the computational effort to solve this eigenvalue problem is usually small, a random distribution of start values in the complex plane between 0 and 1 can be used.

Case where no notch or crack is present

If the notch angle is zero and no crack is present (i.e. all the material interfaces are perfectly bonded), the boundary conditions have to be changed. As a consequence of the continuity of $\boldsymbol{\zeta}$ across material interfaces the solution now has to fulfill:

$$\boldsymbol{\zeta}(\theta_0) = \boldsymbol{\zeta}(\theta_n) \tag{3.18}$$

or equivalently:

$$\begin{bmatrix} \boldsymbol{\zeta}_1(\theta_0) \\ \boldsymbol{p}(\theta_0) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\zeta}_1(\theta_n) \\ \boldsymbol{p}(\theta_n) \end{bmatrix} \tag{3.19}$$

In that case finding a suitable λ from equation (3.14) results in the eigenvalue problem:

$$\det(\boldsymbol{B}(\lambda) - \boldsymbol{I}) = 0 \tag{3.20}$$

Determination of the angular function ψ

It is important to notice, that once λ is known, equation (3.6) can be treated as an ordinary differential equation for the function $\boldsymbol{\psi}(\theta)$. Depending on

the problem one can either construct the vector $\boldsymbol{\psi}(\theta_n)$ in equation (3.17) or one can obtain an eigenvector of equation (3.20). Each of these vectors may directly be used as an initial value for the solution of equation (3.6). For the solution of the resulting initial value problem any ODE-solver can be used.

Determination of the enrichment functions by coordinate transformation

Combining the eigenvalue λ and the solution of equation (3.6) in equation (3.5), a solution of the linear elasticity problem is at hand. The first two components of $\boldsymbol{\zeta}$ are the ones describing the displacement in the plane. To obtain the displacements in cartesian coordinates, they have to be transformed:

$$\begin{bmatrix} u_x(r,\theta) \\ u_y(r,\theta) \end{bmatrix} = \begin{bmatrix} \sin(\theta) & \cos(\theta) \\ -\cos(\theta) & \sin(\theta) \end{bmatrix} \begin{bmatrix} u_\theta(r,\theta) \\ u_r(r,\theta) \end{bmatrix} \tag{3.21}$$

Please note that this transformation should be applied only to the angular components of u_θ and u_r, since r^λ appears in both components and therefore remains unchanged in the transformation. This simplifies the implementation, since otherwise the transformation would have to be calculated at every Gauss point.
Each one of the displacement components obtained from this procedure can be used as an enrichment function. Although the explicit form of the angular part is unknown, the function values can be obtained approximately by linear interpolation using the values calculated by the ODE-solver. The derivatives can be calculated using numerical differentiation.

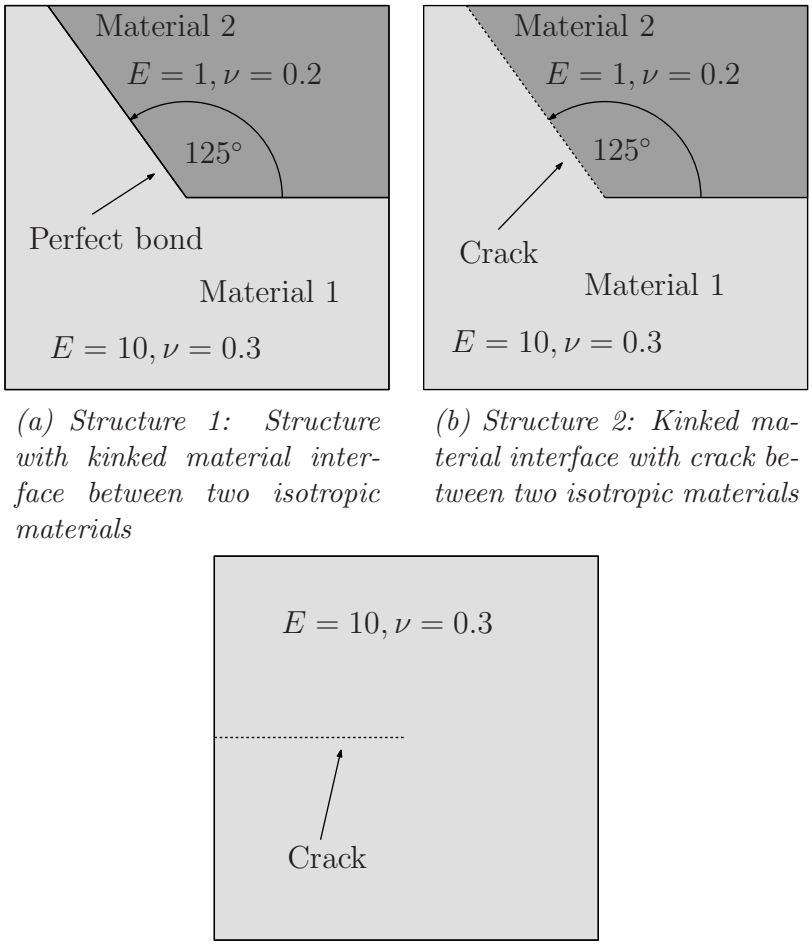
3.2.2. Numerical experiments

First we will justify a choice for the discretization parameter m (equation (3.11)) which will be used throughout this thesis. To do this, numerically determined asymptotic fields are compared to exact solutions. Afterwards numerically determined enrichment functions are applied to different problems in linear elasticity. The results of the X-FEM and the FEM applied to these problems are compared afterwards on several meshes to obtain

convergence rates.

Convergence of the asymptotic behavior of the enrichment functions

To test whether the determination of the enrichment functions described above results in approximations that represent the asymptotic displacement fields of a certain structure, we consider three examples. The first structure



(a) Structure 1: Structure with kinked material interface between two isotropic materials

(b) Structure 2: Kinked material interface with crack between two isotropic materials

(c) Structure 3: Crack in isotropic material

Figure 3.4.: *Different structures with known asymptotic fields used to test the enrichment procedure*

can be seen in figure 3.4(a). Two isotropic materials with different Young's modulus E and Poisson's ratio ν are bonded together. The material interface separating the two materials represents a perfect bond. In the center

	Structure 1	Structure 2
λ^{ref}	0.73448	0.61597 ∓ 0.068874i
$\lambda, m = 25$	$0.74096 - 4.38775$e–11i	0.61819 ∓ 0.069630i
$\lambda, m = 50$	$0.73609 - 9.03411$e–11i	0.61652 ∓ 0.069071i
$\lambda, m = 75$	$0.73519 - 3.29964$e–11i	0.61621 ∓ 0.068962i
$\lambda, m = 100$	$0.73488 - 1.10312$e–11i	0.61610 ∓ 0.068923i
$\lambda, m = 125$	$0.73473 + 3.23075$e–11i	0.61605 ∓ 0.068905i
$\lambda, m = 150$	$0.73465 + 4.53164$e–11i	0.61603 ∓ 0.068896i
$\lambda, m = 175$	$0.73461 + 2.45964$e–11i	0.61601 ∓ 0.068890i
$\lambda, m = 200$	$0.73458 + 5.44040$e–12i	0.61600 ∓ 0.068886i
$\lambda, m = 225$	$0.73456 - 2.20928$e–11i	0.61599 ∓ 0.068884i
$\lambda, m = 300$	$0.73452 - 1.90981$e–11i	0.61598 ∓ 0.068879i

Table 3.2.: *Different discretizations used to calculate* λ

of the structure the interface is kinked and a 125^o angle is formed. Since the Young's modulus is ten times higher in the first material, singularities might occur in the center when the structure is loaded.
The second structure in figure 3.4(b) is the same as the first one, but the material interface is partially cracked. Also in this case singularities can be expected under certain loading conditions.
Vroonhoven [27] states that for these type of problems the displacements in the center of the structure behave asymptotically like r^{λ}. He derives equations from which λ can be determined for different material parameters and angles in the case of a crack as well as in the case of a perfectly bonded interface. We will use the solution obtained from these equations as a reference solution labeled λ^{ref}. The solution procedure described previously should be able to determine a good approximation to λ^{ref}.
The quality of the approximation depends mainly on the number of intervals m chosen for the discretization of each material. The approximation λ for different levels of discretization is shown in table 3.2. Because it is not known a-priori whether the exact solution is complex or not, complex start values for the Newton method were chosen as discussed previously. Therefore, even if λ^{ref} is real, the numerical result contains a small imaginary part. As expected, the approximation improves when the discretization pa-

rameter m is increased. The three leading digits of λ_{ref} for the first problem are reproduced if each material wedge is discretized by 100 intervals. For the second structure 225 intervals per material are necessary to reproduce the three leading digits.

To test the convergence of the angular part of the enrichment functions, we consider the structure shown in figure 3.4(c). An isotropic material contains a crack whose tip is located in the center of the structure. The asymptotic fields are well-known [16] and the components of the displacement fields are often used in X-FEM to describe cracks in isotropic materials:

$$\begin{aligned} u_x(r,\theta) = &\frac{K_I}{2\mu}\sqrt{\frac{r}{2\mu}}\cos\left(\frac{\theta}{2}\right)\left[\kappa - 1 + 2\sin^2\left(\frac{\theta}{2}\right)\right] \\ &+ \frac{K_{II}}{2\mu}\sqrt{\frac{r}{2\mu}}\sin\left(\frac{\theta}{2}\right)\left[\kappa + 1 + 2\cos^2\left(\frac{\theta}{2}\right)\right] \end{aligned} \tag{3.22}$$

$$\begin{aligned} u_y(r,\theta) = &\frac{K_I}{2\mu}\sqrt{\frac{r}{2\mu}}\sin\left(\frac{\theta}{2}\right)\left[\kappa + 1 - 2\cos^2\left(\frac{\theta}{2}\right)\right] \\ &- \frac{K_{II}}{2\mu}\sqrt{\frac{r}{2\mu}}\cos\left(\frac{\theta}{2}\right)\left[\kappa - 1 - 2\sin^2\left(\frac{\theta}{2}\right)\right] \end{aligned} \tag{3.23}$$

κ is the Kolosov constant and μ is the shear modulus. To obtain a reference solution, the stress intensity factors K_I, K_{II} and the radius r are set to 1. The result is a displacement along a circle of unit radius around the crack tip with components u_x^{ref} and u_y^{ref}. The angular functions calculated by solving equation (3.6) and performing the transformation in equation (3.21) should be able to reproduce the displacement field along this circle for sufficiently large values of m. Of course this is only possible if the ODE-solver calculated a reasonable solution of equation (3.6), but the error introduced by this solver is neglected in the further discussion.

To perform a comparison, the angular functions determined by the ODE-solver were used in a least squares fit to obtain approximations $\tilde{u}_x$ and $\tilde{u}_y$ to the reference solutions u_x^{ref} and u_y^{ref}. The error was evaluated in the L_2-norm for each component. The results are shown in table 3.3. The quality of the approximation can be controlled by the choice of m and the numerical results indicate that for sufficiently large values of m, both displacement components converge. Throughout the remainder of this thesis the value $m = 300$ will be used. We may expect a reasonable representation of the

	$\|\tilde{u}_x - u_x^{ref}\|_{L_2}/\|u_x^{ref}\|_{L_2}$	$\|\tilde{u}_y - u_y^{ref}\|_{L_2}/\|u_x^{ref}\|_{L_2}$
$m = 10$	0.01286	0.01677
$m = 20$	0.00353	0.00461
$m = 30$	0.00162	0.00212
$m = 40$	0.00095	0.00124
$m = 50$	0.00065	0.00084
$m = 300$	0.00002	0.00004

Table 3.3.: *Different levels of discretization used to approximate the angular part of the asymptotic displacement fields in structure 3*

asymptotic fields by the enrichment functions for this choice with regard to the results obtained so far.

Convergence for several problems in linear elasticity

Three linear elasticity problems have been chosen to demonstrate the effect of the numerically determined enrichment functions on the convergence rate.

The numerical error $\|e\|_{L_2}$ of a method was evaluated in terms of the displacements in the L_2-norm:

$$\|e\|_{L_2} := \frac{\sqrt{(\int_\Omega \|\boldsymbol{u}^{ref} - \boldsymbol{u}^h\|^2 da)}}{\sqrt{(\int_\Omega \|\boldsymbol{u}^{ref}\|^2 da)}} \tag{3.24}$$

$\boldsymbol{u}^h$ is the numerical solution and $\boldsymbol{u}^{ref}$ is a reference solution. The reference solutions that were used here are either exact solutions or results obtained by the FEM on a very fine mesh.

For both methods the meshes were chosen in the same manner as they were chosen in figures 2.4(a)-2.4(c): A set of equally sized regular squares covers the structure. Each square is then divided in two triangular elements.

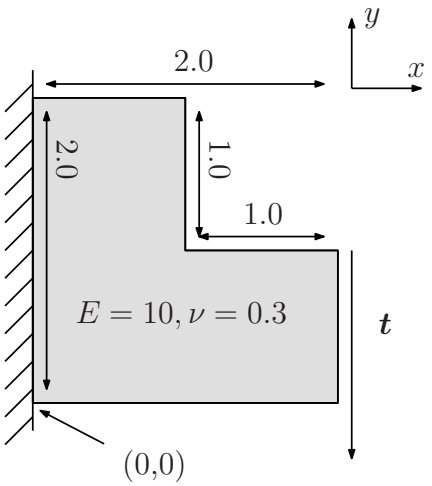
It was assumed that the error can be related to the degrees of freedom (DOFs) by:

$$\|e\|_{L_2} \approx c(\text{DOFs})^l \tag{3.25}$$

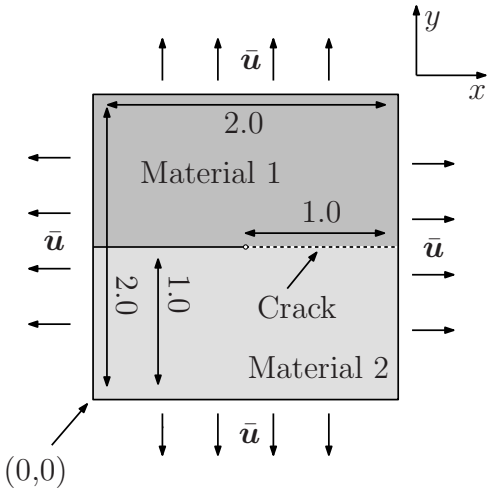
c is a positive constant. l can be determined approximately if the error is calculated for a series of refined meshes. In the following numerical experiments the two finest meshes will be used for this.
For piecewise linear shape functions used in the FEM $l = -1$ is known as the optimal convergence rate. Since the number of degrees of freedom grows cubically with the inverse of the longest element edge h, $l = -1$ would correspond to an error bound of the form ch^2 (we already guaranteed this for the FEM and functions with components in $H^2(\Omega)$, cf. equation (2.56)). However, due to the presence of weak singularities optimal convergence cannot be expected if the FEM is used.
By relating the error to the degrees of freedom we are able to compare the FEM with the X-FEM, since the computational effort for solving the equation systems is roughly the same. However, additional effort has to be made in the X-FEM to calculate the enrichment functions numerically. But this effort is neglected in the comparison.
There is no general rule on how to choose the enrichment radius r_{max}. Reasonable choices were made such that the enriched area always includes several elements but is still small compared to the rest of the structure. The relation of the enrichment radius to the convergence rate was further discussed by Béchet et al. [10]. Based on this discussion we might expect convergence rates close to the optimal rate of $l = -1$ if the non-smooth behavior of the exact solution is represented adequately by the enrichments.

L-shaped structure The first structure is an L-shaped structure as shown in figure 3.5(a). Zero displacements are prescribed at the boundary at $x = 0$. Constant tractions are applied along the boundary at $x = 2$ in the negative y-direction. Due to these boundary conditions a singular stress distribution is formed at the notch.
In the X-FEM the notch at $(1.0, 1.0)$ was enriched with functions obtained by the previously explained procedure. The enrichment radius was chosen as $r_{max} = 0.3$. One eigenvalue of equation (3.16) was found:

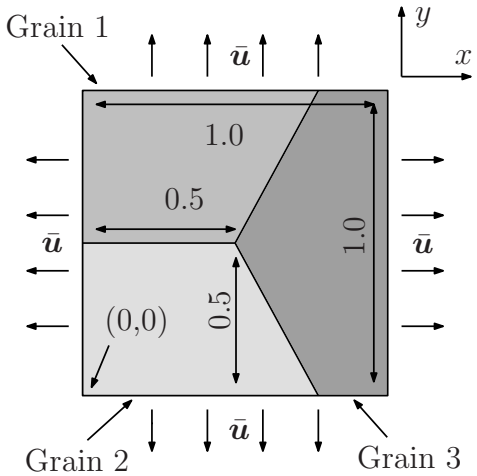
$$\lambda = 0.5445 \tag{3.26}$$



(a) Structure 1: Clamped L-shaped structure with traction forces at the boundary



(b) Structure 2: Bi-material structure with prescribed displacements at the boundary



(c) Structure 3: Polycrystalline structure with prescribed displacements at the boundary

Figure 3.5.: *Different structures used to compare the convergence rate of X-FEM and FEM for linear elasticity problems*

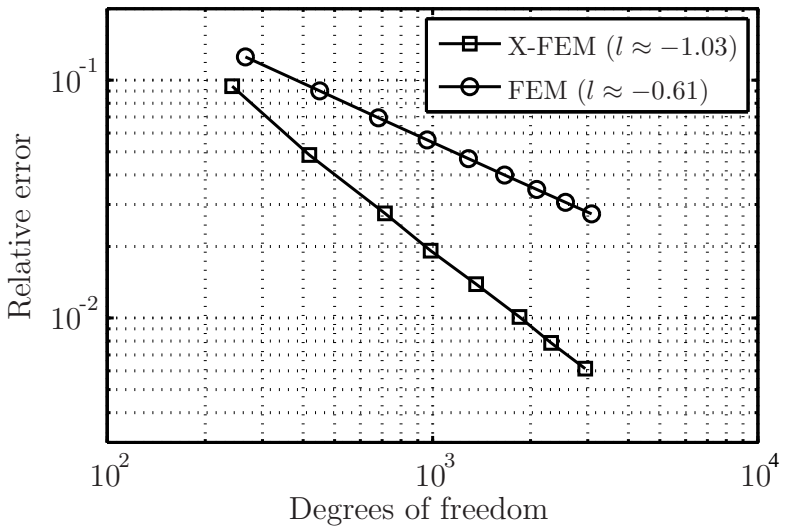
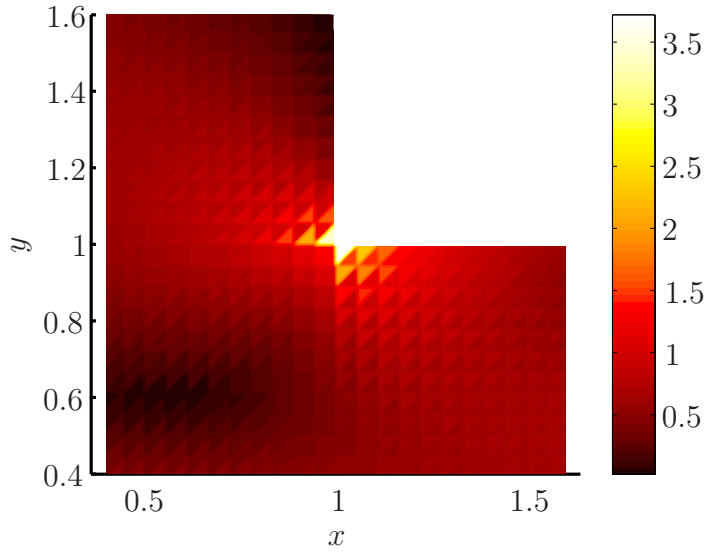


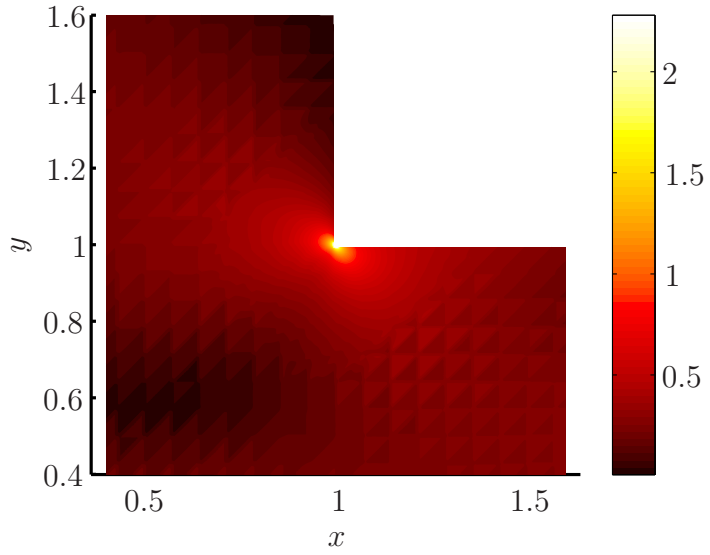
Figure 3.6.: *Displacement error in the L_2-norm for the L-shaped domain*

The multiplicity of this eigenvalue was 1, therefore, two additional enrichments were used in the X-FEM calculation.
A reference solution for this problem was calculated by the FEM on a very fine mesh (375000 DOFs). The relative error for different discretizations is shown in figure 3.6. The error is in general lower if enrichments are used. The convergence rate for the X-FEM is higher than the one for the FEM. The X-FEM convergence rate is as expected close to $l = -1$, from which we can conclude that the singularity is represented very well by the enrichments.
The von Mises stresses are shown in figure 3.7(a) and 3.7(b). The singularity is resolved much better if enrichments are used. In particular we note that X-FEM provides a smoother stress distribution, although less degrees of freedom have been used. Figure 3.6 shows that this stress distribution is associated with a higher accuracy. The stress distribution calculated with the FEM contains large discontinuities at the element edges. Those stress jumps indicate that the exact solution is not represented very well.

Bi-material crack between orthotropic materials The second structure is shown in figure 3.5(b). The two materials are orthotropic. Their interface is partially cracked. A displacement $\bar{\boldsymbol{u}}$ is prescribed at the boundary.



(a) von Mises stress distribution using FEM (2090 DOFs)



(b) von Mises stress distribution using X-FEM (1850 DOFs)

Figure 3.7.: *von Mises stresses for the L-shaped domain*

	Material 1	Material 2
E_x	50	20
E_y	30	10
E_z	10	5
ν_{xy}	0.4	0.3
ν_{yz}	0.3	0.2
ν_{zx}	0.2	0.1
μ_{xy}	5	10

Table 3.4.: *Elastic constants for the orthotropic materials in the bi-material crack problem*

The material parameters characterizing the elastic behavior of the structure are given in table 3.4. In the X-FEM the crack tip at (1.0,1.0) was enriched. Two complex-conjugate solutions of equation (3.16) were found at:

$$\lambda = 0.5 \pm 0.029\text{i} \tag{3.27}$$

Each one of the eigenvalues provides two additional enrichments. Thus, four additional enrichments were used at the crack tip. The asymptotic field associated with the eigenvalue $\lambda = 0.5 + 0.029\text{i}$ was evaluated at the boundary of the structure to obtain displacements $\bar{\boldsymbol{u}}$. The exact solution to the corresponding boundary value problem is then the asymptotic field itself and was used to calculate the error.

To test the convergence rate of the FEM, only meshes were chosen where the element edges conformed to the crack and the interface geometry. To model the crack those element edges simply had to be disconnected. The strain jump along the interface and the displacement jump along the crack can therefore be represented by the shape functions.

To test the convergence rate of the X-FEM, only meshes were chosen where the element edges did not conform to the crack geometry, to stress the fact that this geometric constraint is not an issue in X-FEM. The value for the enrichment radius was again $r_{max} = 0.3$.

The convergence behavior is shown in figure 3.8. Also for this example the use of enrichments improved the convergence rate and decreased the error. While the convergence rate of the X-FEM is close to $l = -1$, the one

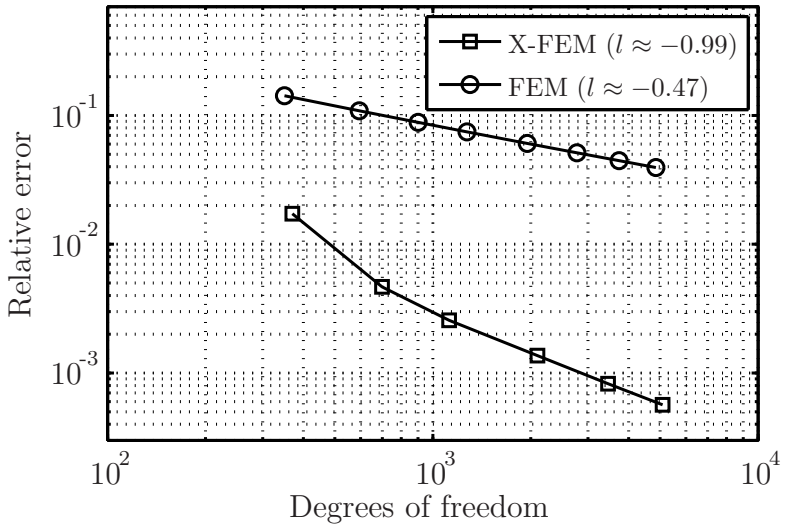
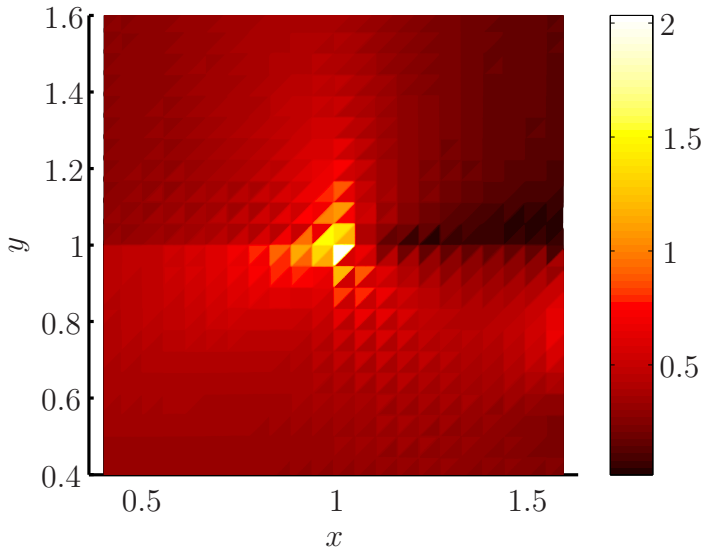


Figure 3.8.: *Displacement error in the L_2-norm for the bi-material crack problem*

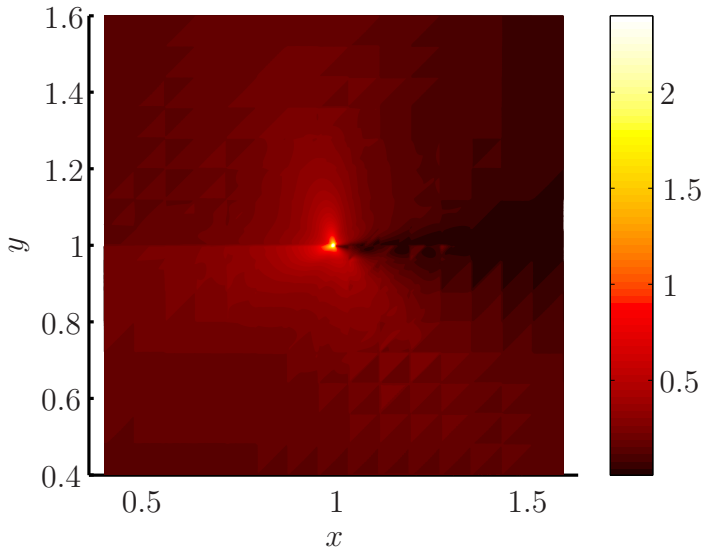
obtained by the FEM is close to $l = -0.5$. It is a well-known result that the convergence rate of the FEM for cracks in isotropic materials is lowered to $l = -0.5$.

The von Mises stress distribution for the two methods can be compared in figure 3.9(a) and figure 3.9(b). A higher number of degrees of freedom was used to generate the stress plot for the FEM. Clearly the stress distribution is not very smooth in that case. This is a result of the inability to reproduce the correct deformation behavior at the crack tip. If X-FEM is used, the problem disappears and a much smoother stress distribution is obtained. The stress jump at the material interface is also reproduced very well by the X-FEM.

Polycrystalline structure The third structure that was used to test the convergence was a polycrystalline structure as shown in figure 3.5(c). Three different grains meet at a common junction, at which their boundaries form angles of 120 degrees. Each of these grains represents a copper crystal. A displacement $\bar{\boldsymbol{u}}$ is prescribed at the boundary of the structure. The elastic constants describing a copper crystal are shown in table 3.5, all other components of the elasticity tensor are zero. This description is only valid if the crystal inside a grain has a certain orientation (i.e. [100]-,[010]- and



(a) von Mises stress distribution using FEM (2774 DOFs)



(b) von Mises stress distribution using X-FEM (2098 DOFs)

Figure 3.9.: *von Mises stresses for the bi-material crack problem*

$D_{1111}, D_{2222}, D_{3333}$	168.4 GPa
$D_{1122}, D_{1133}, D_{2233}, D_{2211}, D_{3311}, D_{3322}$	121.4 GPa
$D_{2323}, D_{3131}, D_{1212}$	75.4 GPa

Table 3.5.: *Elastic constants for a copper grain*

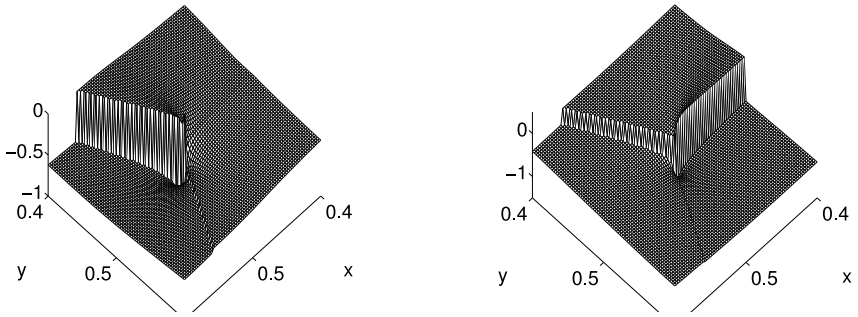
	Grain 1	Grain 2	Grain 3
ϕ_x	0^o	0^o	36^o
ϕ_y	45^o	120^o	0^o
ϕ_z	0^o	-22.5^o	22.5^o

Table 3.6.: *Euler angles describing crystal rotations for the different grains in the polycrystalline structure*

[001]-axis of the crystal align with the x-,y- and z-axes).
For other crystal orientations the entries of the elasticity tensor change. Those can easily be calculated by applying the correct tensor transformation. The transformation is fully described by a rotation matrix, or equivalently by Euler angles ϕ_x, ϕ_y and ϕ_z. The Euler angles for each grain are listed in table 3.6. Copper crystals are known to have an anisotropic behavior. The strain field might contain singularities at the junction in the center due to the different grain orientations. In fact if the junction in the center is considered, equation (3.20) has the solution:

$$\lambda = 0.8738 \tag{3.28}$$

The multiplicity of this eigenvalue was 1, and therefore two additional enrichment functions were obtained from the numerical enrichment procedure. The strains that are formed by the x-component of the asymptotic field are shown in figures 3.10(a) and 3.10(b), the ones that are formed by the y-component in figures 3.10(c) and 3.10(d).
The asymptotic field was evaluated at the boundary of the structure to obtain displacements $\bar{\boldsymbol{u}}$. If these are imposed as boundary conditions the exact solution of the problem is the asymptotic field itself. But in the case of general anisotropy, the asymptotic field has an anti-plane component. To reproduce the asymptotic field the anti-plane component must also be imposed. But imposing such displacements is not possible when standard plane strain elements are used. Therefore, only the x- and the y-component of the asymptotic field were prescribed at the boundary of the structure. But in that case the solution might show a non-smooth behavior at the points where the grain boundaries coincide with the boundary of the structure. This might influence the convergence rate. The problem can be resolved



(a) $\frac{\partial \phi_x}{\partial x}$ *(b)* $\frac{\partial \phi_x}{\partial y}$

(c) $\frac{\partial \phi_y}{\partial x}$ *(d)* $\frac{\partial \phi_y}{\partial y}$

Figure 3.10.: *Strain fields calculated from the enrichment functions used for the polycrystalline structure*

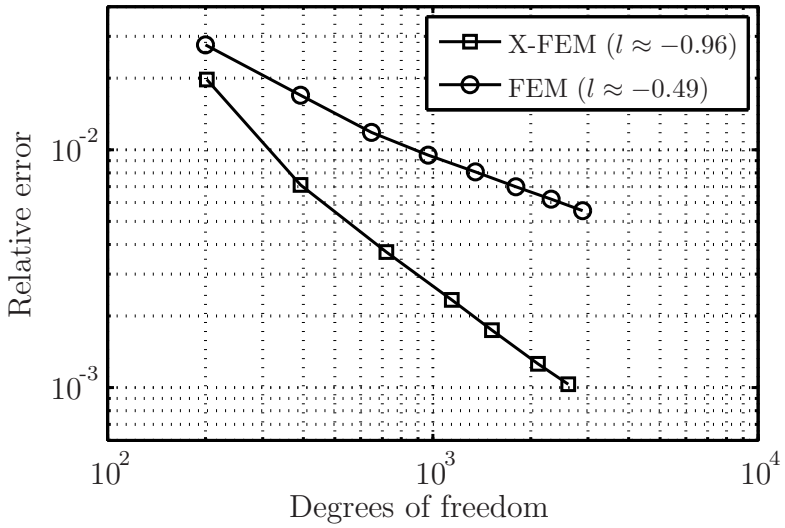


Figure 3.11.: *Displacement error in the L_2-norm for the polycrystalline structure*

by introducing a body force term that eliminates the stresses $\bar{\boldsymbol{\sigma}}$ from the problem which arise in the plane due to the anti-plane component.
The body force term is then singular at the junction in the center and one might argue that such body forces are not very realistic. Therefore, the stresses were multiplied with a cutoff function χ that vanishes in an area around the center of the structure:

$$\chi(r) = \begin{cases} 0 \quad \text{if} \quad r < 0.1 \\ -\frac{1}{2}\left(\cos\left(\pi\frac{r-0.1}{0.3}\right) - 1\right) \quad \text{if} \quad 0.1 \leq r \leq 0.4 \\ 1 \quad \text{if} \quad r > 0.4 \end{cases} \tag{3.29}$$

r is the radial component of the polar coordinate system used to describe the enrichments. The body force term in (2.40) was then chosen as:

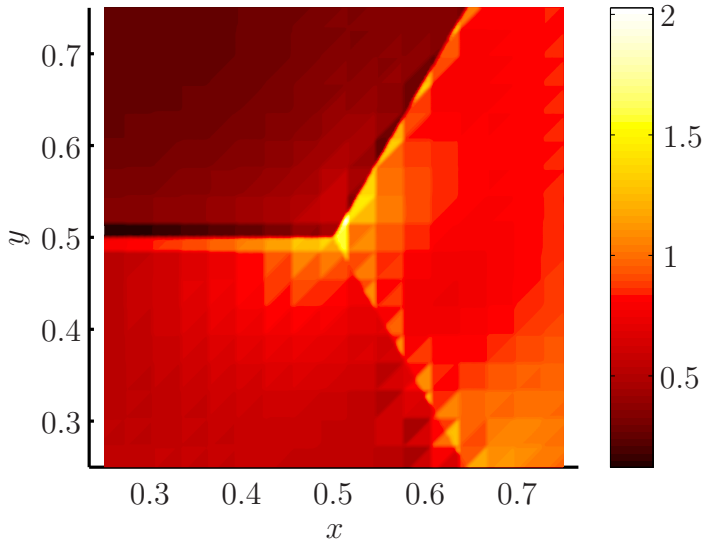
$$\int_{\Omega} (\bar{\boldsymbol{\sigma}}\chi(r)) \, : \, \boldsymbol{\epsilon} da \tag{3.30}$$

A reference solution was calculated for this problem using the FEM (500000 DOFs). The convergence are shown in figure 3.11. The convergence rate achieved by the X-FEM is close to the optimal one, while the FEM performs worse.

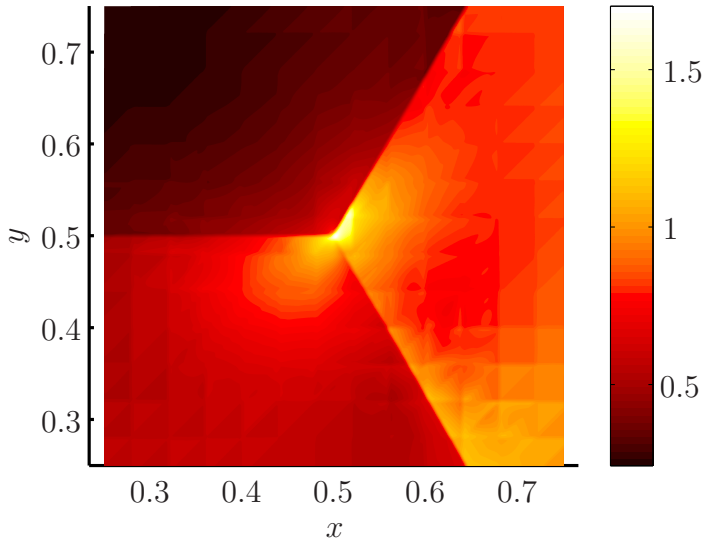
The von Mises stresses for the FEM and the X-FEM are compared in figure 3.12(a) and figure 3.12(b). A visualization of the meshes that have been chosen to calculate these von Mises stresses is shown in figure 3.13(a) and figure 3.13(b). The singularity is resolved very well if X-FEM is used. As in the previous examples, the stress distribution obtained from the FEM calculation contains large discontinuities at the element edges. The stress jumps at the grain boundaries are not reproduced appropriately. Although the grain boundaries do not align with the element edges, those jumps are reproduced by the X-FEM.

3.3. Preconditioning

If the equation systems become large, a direct solution is not efficient any more. Iterative solvers should be used instead. Especially for sparse matrices the solution time can be reduced significantly, since matrix-vector products (which are the basic operations used by iterative solvers, cf. section 2.5) are cheap from a computational point of view. But as discussed previously for the MINRES solver, their performance is highly dependent on the condition number of the stiffness matrix.
For the FEM several criteria to evaluate the quality of a mesh exist. Following these criteria, one can hope for good approximation properties, but also for well-conditioned stiffness matrices. For instance it is known that the condition number grows to infinity if the element size tends to zero. Therefore, one would try to omit extremely small elements in the finite element mesh, at least if they are not necessary to resolve the exact solution appropriately.
For the X-FEM such criteria do not exist. In fact it will be shown that the enrichment schemes explained so far can result in arbitrarily ill-conditioned matrices.
Generally this happens if the enrichments become almost linearly dependent. Béchet et al. [10] proposed a preconditioner specially tailored to the X-FEM, which stabilizes the enrichments by applying Cholesky-decompositions to certain submatrices of the stiffness matrix. These submatrices are formed by the degrees of freedom associated with each enriched node. This can be understood as a local stabilization, because the problem of almost linearly

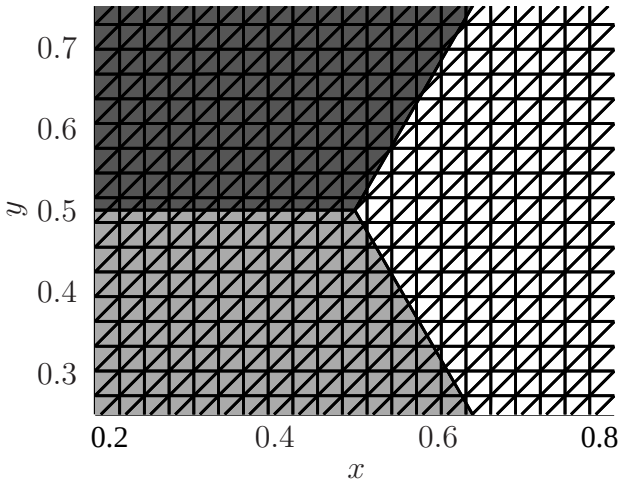


(a) von Mises stress distribution using FEM (2312 DOFs)

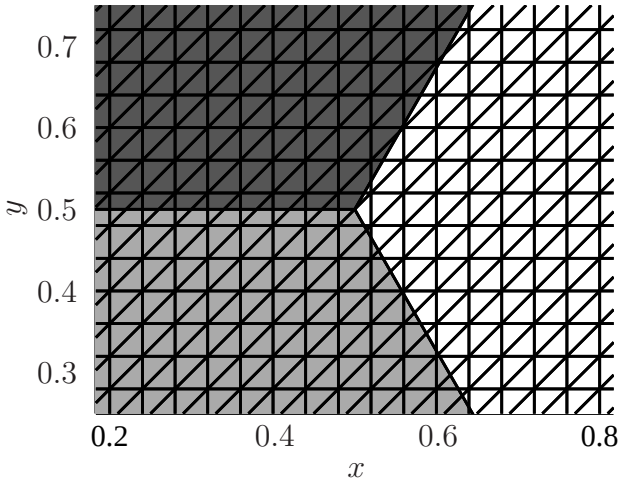


(b) von Mises stress distribution using X-FEM (2106 DOFs)

Figure 3.12.: *von Mises stress distribution for the polycrystalline structure*



(a) Mesh for the FEM (2312 DOFs)



(b) Mesh for the X-FEM (2106 DOFs)

Figure 3.13.: *Meshes used to discretize the polycrystalline structure*

dependent enrichment functions is eliminated for each node. But there are situations in which enrichment functions associated with several nodes become almost linearly dependent. In that case another strategy is needed.
In this section a general technique is described which, applied to the X-FEM, results in well-conditioned stiffness matrices. Their condition number is always close to the one of the corresponding FEM stiffness matrices without any enrichment. The idea was first published and tested by Menk and Bordas [28].
The method can be applied to any sort of enrichment. It is related to the finite element tearing and interconnecting method (FETI) proposed by Farhat and Roux [29]. Both methods employ a domain decomposition. In the FETI-method the subdomains are treated as independent structures. A Cholesky-decomposition is applied to the stiffness matrix associated with each subdomain. Continuity of the solution along the subdomain boundaries is ensured by using additional constraints.
The method proposed here applies the idea of domain decomposition only to the submatrix associated with the enriched degrees of freedom. Similarly as in the FETI-method, Cholesky-decompositions are used together with additional continuity constraints. The method differs from the FETI-method in the way the preconditioner is constructed. To ensure well-conditioned matrices for general enrichments the continuity constraints are transformed and an LQ-decomposition is applied to them.
The FETI-method was used in a different context in connection with X-FEM before by Wyart et al. [30].

3.3.1. Problems in current enrichment schemes

In this section we will discuss one-dimensional problems in which the previously described enrichments can result in extremely ill-conditioned systems. The first example involves a material interface and similar examples are possible in two-dimensional settings.
In the second example a special type of enrichment was chosen to give an explanation of what happens in two dimensions if the enrichment radius around the crack tip is large.
To interpret these examples correctly some linear algebra tools are needed.

By equation (2.112) the equation system is ill-conditioned if the smallest eigenvalue approaches zero. An upper bound for the smallest eigenvalue of a symmetric positive definite matrix $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ is given by the Rayleigh-quotient $q_R(\boldsymbol{a})$ for any non-zero vector $\boldsymbol{a} \in \mathbb{R}^n$:

$$q_R(\boldsymbol{a}) := \frac{\boldsymbol{a}^T \boldsymbol{K} \boldsymbol{a}}{\boldsymbol{a}^T \boldsymbol{a}} \geq \lambda_{min} \geq 0 \tag{3.31}$$

If the matrix stems from a discretization of a one-dimensional structural problem the vector $\boldsymbol{a}$ contains the coefficients for a function u^h from the discrete space S^h. Using the bilinear form from equation (2.17) the Rayleigh-quotient can equivalently be written as:

$$q_R(\boldsymbol{a}) = \frac{a(u^h, u^h)}{\boldsymbol{a}^T \boldsymbol{a}} \geq \lambda_{min} \geq 0 \tag{3.32}$$

Material Interface

The one-dimensional problem is depicted in figure 3.14. A material interface

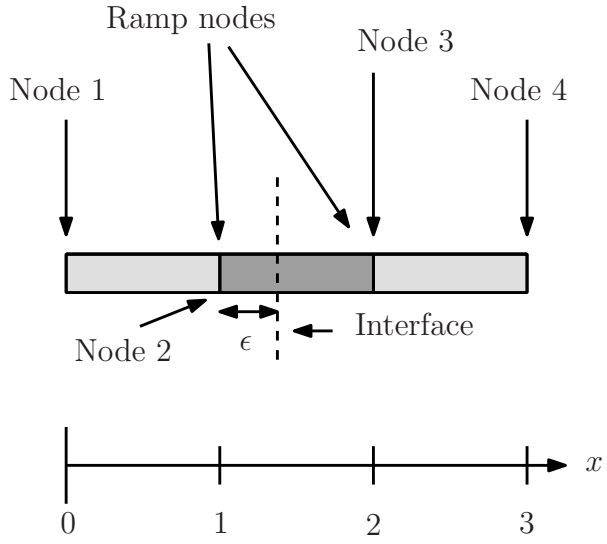


Figure 3.14.: *One-dimensional interface problem*

is at distance ϵ from node 2.

All four nodes are enriched. The enrichment functions developed in section 2.4 for $\epsilon = 0.1$ are shown in figure 3.15. An interesting phenomena can be observed here. The enrichment functions form little "hills" and "valleys". Just by looking at the functions one might guess that they almost cancel

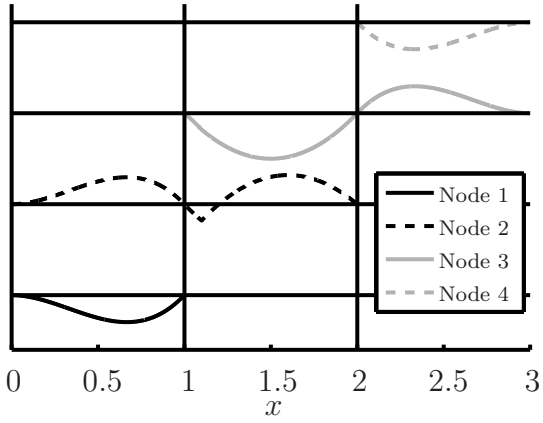


Figure 3.15.: *Nodal enrichments for the material interface with* $\epsilon = 0.1$

out if they are summed up.
This is exactly what happens as we can see in figure 3.16. For smaller

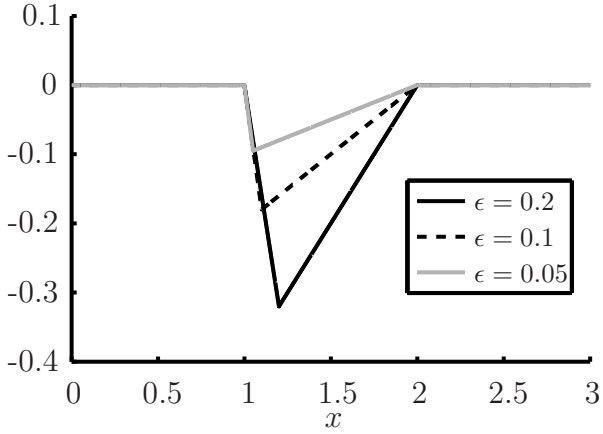


Figure 3.16.: *Sum of the enrichment functions for the material interface problem evaluated for different values of* ϵ

ϵ-values the sum approaches the zero function. We can observe the same effect for the derivative of the sum. This is shown in figure 3.17.
Thus, in our example a linear combination of enrichment functions has a derivative which approaches the zero function for $\epsilon \to 0$. In this one-dimensional case the bilinear form from equation (3.32) is simply:

$$a(u^h, u^h) = \int_\Omega E \left(\frac{du^h}{dx} \right)^2 dx \tag{3.33}$$

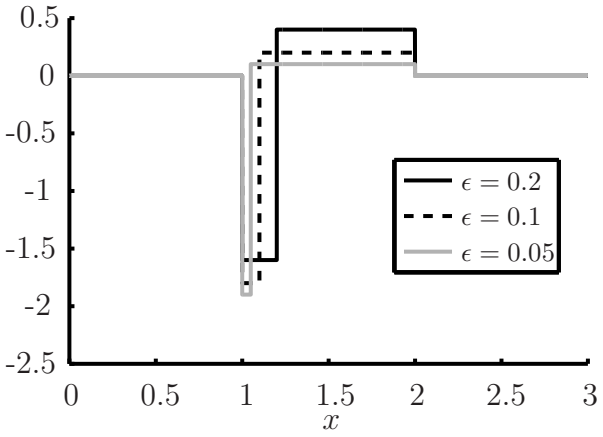


Figure 3.17.: *Sum of the derivative of the enrichment functions for the material interface problem evaluated for different values of* ϵ

E is Young's modulus and Ω is the interval $(0, 3)$. The bilinear form vanishes for this sum if the interface approaches node 2, but the corresponding vector of coefficients $\boldsymbol{a}$ remains unchanged ($\boldsymbol{a}$ contains ones and zeros such that the nodal enrichments are summed up). By (3.32) we can conclude that the smallest eigenvalue of the discretized problem therefore also vanishes. This results in an ill-conditioned equation system.
A corresponding two-dimensional example would be an interface approaching a line of element edges. This problem is investigated in the numerical experiments in section 3.3.3.

Crack tip enrichment

To investigate the situation in the case of weakly singular enrichment functions and large enrichment radii, we first enrich a 1-dimensional problem with a linear function, say $P(x) = x$, using (2.81). The discretized structure is depicted in figure 3.18. The structural domain is discretized by 5 finite elements. The whole domain is enriched and the interior nodes are the ramp nodes.
The nodal enrichments that are obtained by a linear enrichment function are shown in figure 3.19. The situation is similar to the one observed in the case of a material interface. The nodal enrichments form "hills" and "valleys". The "hill" of an enrichment function seems to be similar to the "valley" of the neighboring nodal enrichment, only the sign differs. And in

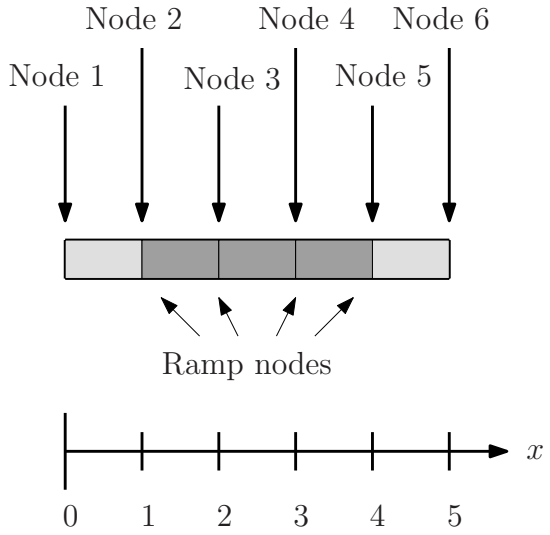


Figure 3.18.: *One-dimensional problem used for the enrichment with a linear function*

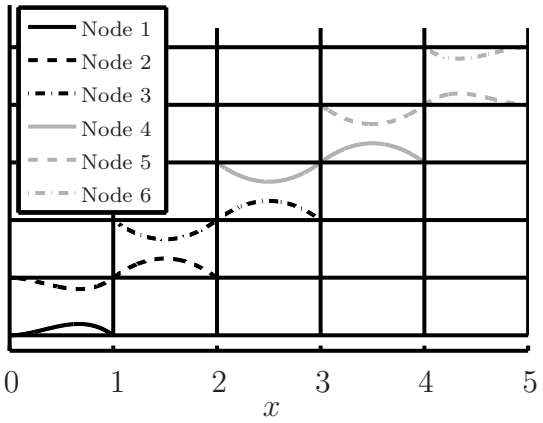


Figure 3.19.: *Nodal enrichments obtained from linear function enrichment*

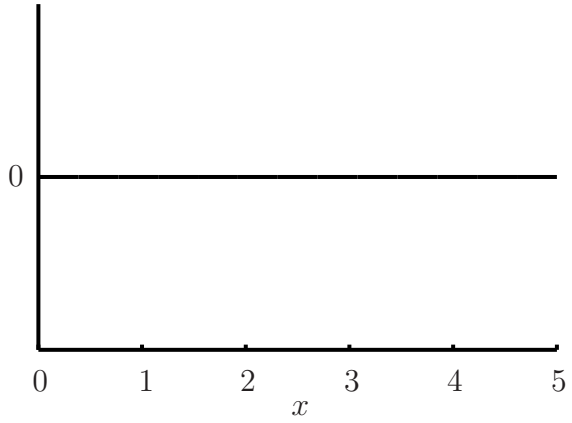


Figure 3.20.: *Sum of nodal enrichments for a linear enrichment function*

fact the sum of all the enrichments is the zero function shown in figure 3.20. Thus, the derivative also vanishes. Therefore, the left hand side of the estimate in equation (3.32) is zero (again by evaluating the expression in equation (3.33)) and so is the smallest eigenvalue of the stiffness matrix.
Of course it is not useful to enrich the FE-space with a linear function, since linear functions can be reproduced by the polynomial shape functions. But there may be situations in the 2-dimensional case, in which the enrichment function is close to a linear function in a certain area. Let us define the following function:

$$CT(x) = \sqrt{r(x)} \cos\left(\frac{\theta(x)}{2}\right) \tag{3.34}$$

$$r(x) = \sqrt{d^2 + x^2} \tag{3.35}$$

$$\theta(x) = \tan^{-1}\left(\frac{x}{d}\right) \tag{3.36}$$

If the arguments θ and r were independent variables, then $CT(r, \theta)$ would be one of the standard crack tip enrichment functions [22]. But here θ and r depend on x in a special manner. Evaluation of $CT(x)$ is an evaluation of that enrichment function along a straight line perpendicular to a crack with distance d to its crack tip. By changing the value d we are able to see how the enrichment function behaves further away from the crack tip. The situation is depicted in figure 3.21.
For different values of d the function $CT(x)$ is shown in figure 3.22. Clearly,

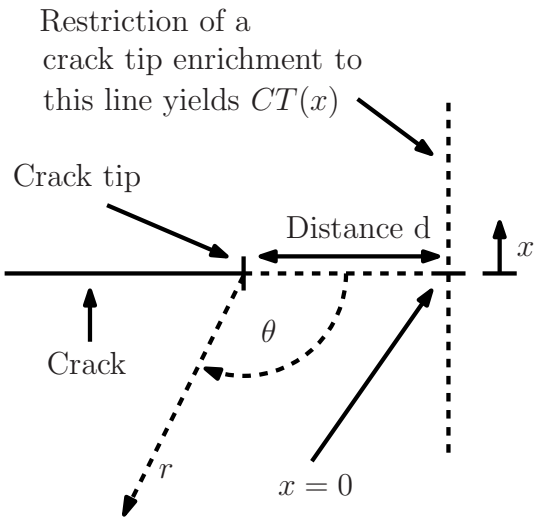


Figure 3.21.: *Evaluation of a crack tip enrichment function along a line in front of the crack tip at distance d*

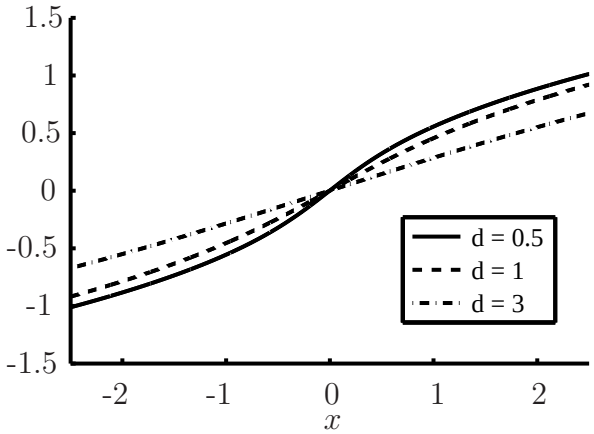


Figure 3.22.: *Evaluation of the function $CT(x)$ for different values of d*

for increasing values of d the function $CT(x)$ is well approximated by a linear function.
We can conclude that in this situation the smallest eigenvalue of the stiffness matrix is close to zero and the system is ill-conditioned. In the 2-dimensional case the nodal enrichments of elements ahead of the crack tip will, at some distance, show a similar behavior as the functions in figure 3.19. The numerical experiments in section 3.3.3 show that in the case of a large enrichment radius the stiffness matrix indeed becomes extremely ill-conditioned.

3.3.2. Calculation of the preconditioner

General form of the preconditioner

Instead of solving the equation system (2.46) we attempt to solve:

$$\underbrace{\boldsymbol{P}^T\boldsymbol{K}\boldsymbol{P}}_{\tilde{\boldsymbol{K}}}\underbrace{\boldsymbol{P}^{-1}\boldsymbol{a}}_{\tilde{\boldsymbol{a}}} = \underbrace{\boldsymbol{P}^T\boldsymbol{f}}_{\tilde{\boldsymbol{f}}} \tag{3.37}$$

Once this equation system is solved to obtain $\tilde{\boldsymbol{a}}$ one can calculate the solution of equation (2.46) simply by:

$$\boldsymbol{a} = \boldsymbol{P}\tilde{\boldsymbol{a}} \tag{3.38}$$

The matrix $\boldsymbol{P}$ is a preconditioner. It should be chosen such that the condition number of the transformed stiffness matrix $\tilde{\boldsymbol{K}}$ is smaller than the one of $\boldsymbol{K}$. Furthermore, the effort necessary for its computation should be small. The advantage is that an iterative solver would need a smaller number of iterations to get an acceptable result. This can justify the additional effort, especially if several solutions for different vectors $\boldsymbol{f}$ have to be calculated. The explicit form of $\tilde{\boldsymbol{K}}$ is not calculated. When iterative solution algorithms are used the explicit form is not needed, in every iteration step it suffices to calculate a matrix-vector product with each of the matrices forming $\tilde{\boldsymbol{K}}$ (cf. section 2.5).

Domain decomposition

Domain decomposition methods are used in many contexts to deal with numerical difficulties encountered in the simulation of complex structures

(cf. [31, 32]). In this thesis domain decompositions will be used to develop a robust preconditioner for the X-FEM.

The problems described in section 3.3.1 are a result of the enrichment procedure. They could be resolved if a matrix decomposition, such as the Cholesky-decomposition, is applied to the submatrix formed by the enriched degrees of freedom. The inverse of the Cholesky-factor could then be used to form a preconditioner. The disadvantage of such matrix decomposition algorithms is that the computation time as well as the memory consumption depends cubically on the matrix size.

Although in most applications the number of enriched degrees of freedom is much smaller than the number of standard degrees of freedom, there are a lot of situations in which the number of enriched degrees of freedom is still very large (e.g. polycrystalline structures [15], multiple cracks [33] and geometric enrichment [10]).

If the structure would be decomposed into several smaller disconnected domains, the submatrix associated with the enriched degrees of freedom would be a block diagonal matrix. In that case a Cholesky-decomposition has to be applied only to each one of these smaller blocks. This would decrease the numerical effort. But how to apply this idea to a structure that is actually connected?

In that case the structure has to be split into several subdomains and continuity conditions must be added. To resolve the problems encountered in section 3.3.1, it suffices to apply this idea only to the enrichment functions. In the following a possible implementation is described more precisely.

The degrees of freedom of the X-FEM stiffness matrix can be ordered such that:

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}_{FEM,FEM} & \boldsymbol{K}_{X,FEM} \\ \boldsymbol{K}_{FEM,X} & \boldsymbol{K}_{X,X} \end{bmatrix} \tag{3.39}$$

$\boldsymbol{K}_{FEM,FEM}$ is the submatrix formed by the standard degrees of freedom and $\boldsymbol{K}_{X,X}$ the one formed by the enriched degrees of freedom.

Let us assume that the domain Ω is decomposed into several non-overlapping subdomains Ω_i. Each subdomain is formed by a union of elements. The matrix $\boldsymbol{K}_{X,X}$ is evaluated as if these domains were disconnected. The matrix $\boldsymbol{K}_{FEM,FEM}$ is evaluated as usual, that is, treating the domain Ω as

one connected domain. An example is given in figure 3.23. A domain is

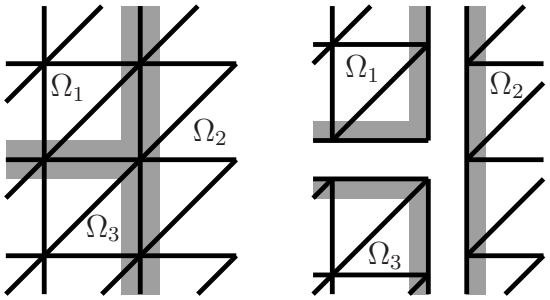


Figure 3.23.: *Example of a domain decomposition, the nodal enrichments are split in the domain decomposition, the FEM shape functions remain unchanged*

decomposed into three domains. Each of the subdomains is formed by a union of elements. The number of standard degrees of freedom does not change, since the decomposition has no effect on the corresponding part of the stiffness matrix. The number of enriched degrees of freedom increases due to the nodes at the boundary of the subdomains.
Since the domains are disconnected, the submatrix of enriched degrees of freedom can be written as a block-diagonal matrix:

$$\boldsymbol{K}_{X,X} = \begin{bmatrix} \boldsymbol{K}_{X,X}^{\Omega_1} & \mathbf{0} & \\ \mathbf{0} & \boldsymbol{K}_{X,X}^{\Omega_2} & \\ & & \ddots \end{bmatrix} \tag{3.40}$$

$\boldsymbol{K}_{X,X}^{\Omega_i}$ are the enriched degrees of freedom of all the enrichments that are non-zero inside the domain Ω_i.
But without further restrictions on the function space this new system would give an erroneous solution, since the enrichment functions might be discontinuous at the boundaries of the subdomains. We therefore use an additional matrix $\boldsymbol{Y}$ to ensure continuity of the displacements (at least in the absence

of cracks):

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}_{FEM,FEM} & \boldsymbol{K}_{X,FEM} & \boldsymbol{0} \\ \boldsymbol{K}_{FEM,X} & \boldsymbol{K}_{X,X} & \boldsymbol{Y}^T \\ \boldsymbol{0} & \boldsymbol{Y} & \boldsymbol{0} \end{bmatrix} \tag{3.41}$$

To demonstrate how $\boldsymbol{Y}$ is constructed we consider again figure 3.23. Assume that the i-th node is at the boundary of two subdomains (e.g. it is connected to Ω_1 and Ω_2 but not Ω_3). Also assume that there is one enriched degree of freedom associated with this node in the standard X-FEM formulation described in section 2.4. Then due to the domain decomposition there would be two enriched degrees of freedom associated with this node in the stiffness matrix in equation (3.41), one associated with Ω_1, the other one with Ω_2. If the vector $\boldsymbol{a}^{X_i}$ contains those degrees of freedom, then a condition for the continuity of the nodal enrichment can be written as:

$$\begin{bmatrix} 1 & -1 \end{bmatrix} \boldsymbol{a}^{X_i} = 0 \tag{3.42}$$

If the node is the one in figure 3.23 which is connected to all three domains, then the condition becomes:

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix} \boldsymbol{a}^{X_i} = 0 \tag{3.43}$$

The whole matrix $\boldsymbol{Y}$ is then constructed by ensuring such conditions for all enriched nodes that are connected to two or more subdomains. More precisely, for every nodal enrichment whose support is contained in n distinct subdomains $n-1$ rows must be added to the matrix $\boldsymbol{Y}$. They are constructed using the $(n-1) \times n$ matrix:

$$\begin{bmatrix} 1 & -1 & 0 & & \cdots & 0 \\ 0 & 1 & -1 & 0 & \cdots & 0 \\ & & & \vdots & & \\ 0 & \cdots & 0 & 1 & -1 & 0 \\ 0 & \cdots & & 0 & 1 & -1 \end{bmatrix} \tag{3.44}$$

To construct the new rows of $\boldsymbol{Y}$ the columns of (3.44) are placed in an empty matrix. Their position is determined by the degrees of freedom that are assigned to the nodal enrichment after performing the domain decomposition.

The new preconditioner

Each block $\boldsymbol{K}_{X,X}^{\Omega_i}$ in (3.40) can be decomposed by a Cholesky-decomposition with Cholesky factor $\boldsymbol{C}_i$:

$$\boldsymbol{K}_{X,X}^{\Omega_i} = \boldsymbol{C}_i{}^T \boldsymbol{C}_i \tag{3.45}$$

We define the preconditioner for $\boldsymbol{K}_{X,X}$ as:

$$\boldsymbol{P}_X = \begin{bmatrix} \boldsymbol{C}_1^{-1} & \boldsymbol{0} & \\ \boldsymbol{0} & \boldsymbol{C}_2^{-1} & \\ & & \ddots \end{bmatrix} \tag{3.46}$$

This preconditioner transforms $\boldsymbol{K}_{X,X}$ to the unit matrix and therefore would remove the small eigenvalues:

$$\boldsymbol{P}_X^T \boldsymbol{K}_{X,X} \boldsymbol{P}_X = \boldsymbol{I} \tag{3.47}$$

Please note, that the inverse of the Cholesky-factors should not be calculated. It suffices to provide a routine for the iterative solver that calculates the matrix-vector product with $\boldsymbol{P}_X^T$ and $\boldsymbol{P}_X$. Therefore, in each iteration step several small equation systems with the Cholesky-factors can be solved instead by performing forward and backward substitution.
A preconditioner for the whole system could then be:

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{P}_{FEM} & \boldsymbol{0} & \\ \boldsymbol{0} & \boldsymbol{P}_X & \boldsymbol{0} \\ & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \tag{3.48}$$

$\boldsymbol{P}_{FEM}$ can be any preconditioner for the standard degrees of freedom. The final equation system may however still be ill-conditioned because $\tilde{\boldsymbol{K}}$ contains the matrix $\boldsymbol{Y}\boldsymbol{P}_X$ whose rows can be almost linearly dependent. Applying

an LQ-decomposition to $\boldsymbol{Y}\boldsymbol{P}_X$ we see that:

$$\boldsymbol{Q} = \boldsymbol{L}^{-1}\boldsymbol{Y}\boldsymbol{P}_X \tag{3.49}$$

$\boldsymbol{L}$ is a lower triangular matrix and $\boldsymbol{Q}$ is a matrix with orthogonal rows. The final preconditioner then takes the form:

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{P}_{FEM} & \boldsymbol{0} & \\ \boldsymbol{0} & \boldsymbol{P}_X & \boldsymbol{0} \\ & \boldsymbol{0} & \boldsymbol{L}^{-1} \end{bmatrix} \tag{3.50}$$

Thus, the transformed equation system becomes:

$$\tilde{\boldsymbol{K}} = \begin{bmatrix} \tilde{\boldsymbol{K}}_{FEM,FEM} & \tilde{\boldsymbol{K}}_{X,FEM} & \boldsymbol{0} \\ \tilde{\boldsymbol{K}}_{FEM,X} & \boldsymbol{I} & \boldsymbol{Q}^T \\ \boldsymbol{0} & \boldsymbol{Q} & \boldsymbol{0} \end{bmatrix} \tag{3.51}$$

A comment must be made about the Cholesky decomposition in equation (3.45). When using domain decomposition in combination with the FEM the stiffness matrix of a subdomain may become indefinite if appropriate boundary conditions are missing for this particular subdomain. In the absence of displacement boundary conditions the substructures are able to perform rigid body motions which results in zero eigenvalues. The Cholesky-decomposition is likely to fail in that case.
However, in the examples considered here the matrices $\boldsymbol{K}_{X,X}^{\Omega_i}$ in (3.45) have always been positive definite. The reason is that nodal shifting is applied to the enrichments as described in section 2.4. Due to this nodal shifting the enrichments are zero at all nodes. Thus, it is not possible to describe rigid body motions using the enrichments.
But in certain situations zero eigenvalues still may occur. Consider for instance figure 3.15. If the domain is decomposed into two domains at $x = 1$, the enrichment functions of node 1 and node 2 in the left domain would be linearly dependent. In that case the eigenvectors corresponding to the zero eigenvalues must be identified (e.g. by using a Krylov-type method).
Assume that the columns of $\boldsymbol{Z}$ contain an orthogonal basis of the null space.

A simple way to deal with the problem of indefiniteness would be to perform the Cholesky-decomposition for a stabilized version of $\boldsymbol{K}_{X,X}^{\Omega_i}$:

$$\boldsymbol{K}_{X,X}^{\Omega_i} + \lambda \boldsymbol{Z}\boldsymbol{Z}^T \quad \text{with} \quad \lambda > 0 \tag{3.52}$$

In an algorithm one would always try to do a Cholesky-decomposition first. If the decomposition fails or if extremely small entries appear on the diagonal, the matrix $\boldsymbol{Z}$ must be determined.
However, such cases were not encountered in the calculations performed in this thesis and therefore this aspect is neglected in the further discussion.

Relationship to the FETI-method

The preconditioning technique was inspired by the FETI-method [29]. In the following it is explained what modifications have to be made to obtain the FETI-method in its original form. First we will assume that all degrees of freedom are labeled as enriched. This enables us to reuse the previously introduced notation. The equation system then becomes:

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}_{X,X} & \boldsymbol{Y}^T \\ \boldsymbol{Y} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{a}^X \\ \boldsymbol{a}^Y \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{0} \end{bmatrix} \tag{3.53}$$

To simplify the explanation we will assume that displacement boundary conditions are prescribed for each subdomain such that each $\boldsymbol{K}_{X,X}^{\Omega_i}$ is non-singular. The main part of the FETI-method consists of solving:

$$\boldsymbol{Y}\boldsymbol{K}_{X,X}^{-1}\boldsymbol{Y}^T\boldsymbol{a}^Y = \boldsymbol{Y}\boldsymbol{K}_{X,X}^{-1}\boldsymbol{f} \tag{3.54}$$

The equation system is solved iteratively using the preconditioner:

$$\boldsymbol{P}_{FETI} = \boldsymbol{Y}\boldsymbol{K}_{X,X}\boldsymbol{Y}^T \tag{3.55}$$

The equation system (3.54) is multiplied only from the left with this preconditioner.
The missing part $\boldsymbol{a}^X$ of the solution is calculated afterwards by solving a small problem on each subdomain. For FEM discretizations this results in well-conditioned equation systems if an appropriate domain decomposition

is used. However, in the case of a general enrichment this cannot be guaranteed.
The difference to the FETI-method can be summarized as follows:

- Domain decomposition is applied only to the enriched degrees of freedom;
- The matrix $\boldsymbol{Y}$ is transformed using the Cholesky-decompositions;
- An LQ-decomposition is applied to the transformed matrix $\boldsymbol{Y}\boldsymbol{P}_X$;
- The resulting preconditioner is applied from the left and from the right to the stiffness matrix.

A comment on the computational effort

The main part of the computational effort in section 3.3.2 consists of computing the Cholesky-decompositions for each subdomain and the LQ-decomposition of $\boldsymbol{Y}\boldsymbol{P}_X$. Evaluating $\boldsymbol{Y}\boldsymbol{P}_X$ itself involves solving small equation systems with the Cholesky-factors. But the sparsity of $\boldsymbol{Y}$ can be exploited such that the calculation of $\boldsymbol{Y}\boldsymbol{P}_X$ is negligible when the overall computational effort is addressed.
A Cholesky-decomposition of a fully populated matrix $\boldsymbol{K}_{X,X}^{\Omega_i} \in \mathbb{R}^{n\times n}$ takes $\frac{n^3}{3}$ [34] floating point operations (flops). An LQ-decomposition of a fully populated matrix $\boldsymbol{Y}\boldsymbol{P}_X \in \mathbb{R}^{n\times m}$ with $m > n$ takes about $3n^2m$ [34] flops.
The matrices that have to be decomposed are usually sparse matrices. Therefore, one should apply sparse versions of the matrix decomposition algorithms. But the estimates given above will serve as a rule of thumb to evaluate a certain domain decomposition in the numerical experiments in section 3.3.3.
The effort for the matrix decompositions can be kept small in most X-FEM applications because:

- The matrix decompositions are only applied to enriched degrees of freedom. The number of enriched degrees of freedom in most industry applications is much smaller than the number of standard degrees of freedom.

- By choosing an appropriate domain decomposition the overall effort for the matrix decompositions can be minimized.
  A large subdomain with many enriched degrees of freedom will result in a computationally expensive Cholesky-decomposition. A large number of subdomains will result in a computationally expensive LQ-decomposition.
  If the number of enriched degrees of freedom is roughly the same for each subdomain, the same holds for each Cholesky-decomposition. The number of subdomains can be chosen such that the effort for the LQ-decomposition is balanced against the average effort for the Cholesky-decompositions.

- The computation of the Cholesky-decompositions for each subdomain can be done simultaneously on different processors. $\boldsymbol{P}_X$ can therefore be computed fully in parallel. The same is true for the matrix-vector product with $\boldsymbol{P}_X$ and its transpose that has to be evaluated in each iteration step.

3.3.3. Numerical experiments

The effect of the preconditioner on the condition number was evaluated using two structures which can be interpreted as two-dimensional generalizations of the problems discussed in section 3.3.1. A third structure containing multiple cracks was used to address the performance in more practical situations.
The following Matlab-routines were used in the experiments: `chol()` was used to calculate the Cholesky-decompositions and `qr()` to calculate the LQ-decomposition. `minres()` is an implementation of the MINRES iterative solver and was used to solve the equation systems. Forward and backward substitution to solve triangular equation systems was done using `mldivide()`. The condition number was evaluated using `eigs()`.
The presentation of each example will be followed by a short discussion about the computational effort. In section 3.3.3 and 3.3.3 this discussion will only be based on the simple rules developed in section 3.3.2. The purpose is to give an idea about how the domain decomposition affects the computational effort.

The example in section 3.3.3 is considered to be more closely related to standard applications of X-FEM. To address the usefulness of the preconditioner actual computation times are presented and discussed.
The preconditioner for the standard degrees of freedom was chosen as:

$$\boldsymbol{P}_{FEM} = \sqrt{\text{diag}(\boldsymbol{K}_{FEM,FEM})^{-1}} \tag{3.56}$$

diag(·) denotes the diagonal of a matrix. This is known as the Jacobi preconditioner or as diagonal scaling.
The equation systems obtained from the procedure in section 3.3.2 will subsequently be referred to as the preconditioned version of X-FEM.
The results are compared with those of the standard X-FEM as described in section 2.4. For these equation systems diagonal scaling was applied to the whole stiffness matrix. They will subsequently be referred to as the standard version of X-FEM.

Material interface

The first structure is shown in figure 3.24. It contains a material interface

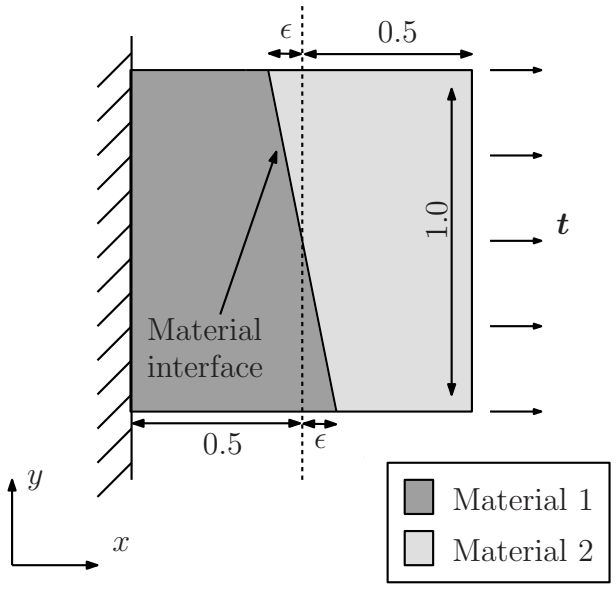


Figure 3.24.: *Structure with slanting material interface*

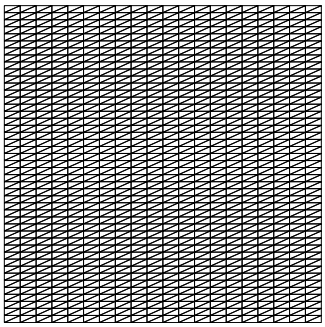
separating material 1 and material 2. A vertical line through the center

of the structure is also shown in the figure. The material interface is not quite vertical, its distance towards the vertical line at the boundary of the structure is characterized by the distance ϵ.
The materials are both isotropic materials with material constants shown in table 3.7. Zero displacements are prescribed at the boundary on the left

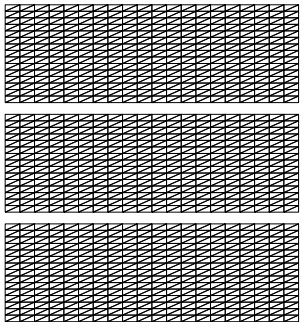
	Material 1	Material 2
E	10	20
ν	0.3	0.3

Table 3.7.: *Material constants for structure with slanting material interface*

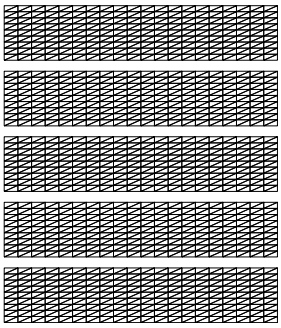
and a constant traction $\boldsymbol{t}$ on the right.
A simple mesh was used to discretize this structure consisting of 1800 equally sized elements. The mesh and different domain decompositions are shown in figure 3.25. The mesh around the center of the structure is shown in figure 3.26. The blending elements and the fully enriched elements are indicated. The condition number of the stiffness matrices has been evaluated for different values of ϵ. The results are shown in figure 3.27. The condition number of the standard finite element matrices (i.e. the diagonally scaled submatrix $\tilde{\boldsymbol{K}}_{FEM,FEM}$) is slightly below the ones obtained for the preconditioned version. For the preconditioned version the choice of domain decomposition as well as the value of ϵ has no effect on the condition number. The condition number of the standard version, however, grows for decreasing values of ϵ. If $\epsilon = 10^{-7}$ the difference between the standard and the preconditioned version is three orders of magnitude.
The relative error of the equation systems after different iteration steps and $\epsilon = 10^{-7}$ is shown in figure 3.28. Clearly, the solver converges much faster if the preconditioned version is used. The difference between the standard and the preconditioned version is due to the different condition numbers observed in figure 3.27.
To conclude this example we analyze the numerical effort based on the values given in table 3.8. The number of enriched degrees of freedom increases slightly for the preconditioned version. If the domain is decomposed into 5 subdomains the number of enriched degrees of freedom is higher because



(a) Mesh used to discretize the material interface problem



(b) Domain decomposed into 3 subdomains



(c) Domain decomposed into 5 subdomains

Figure 3.25.: *Mesh used to discretize the material interface problem and different domain decompositions*

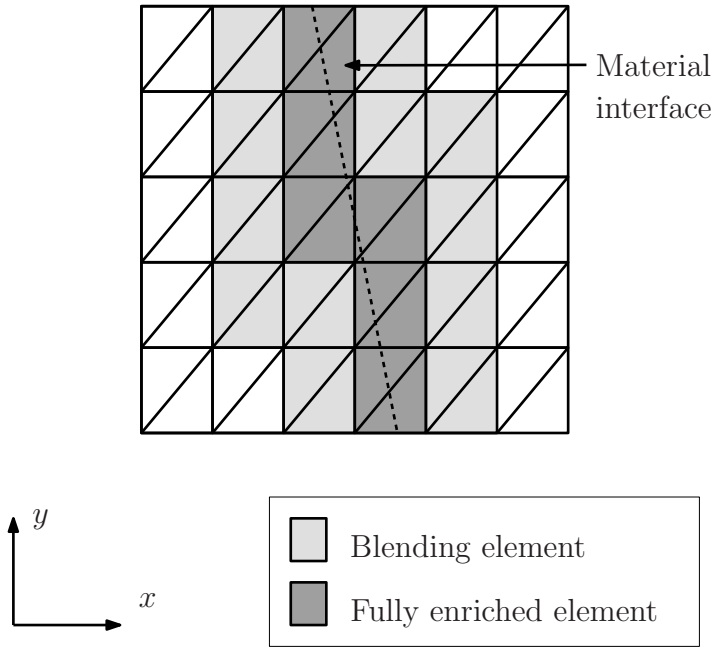


Figure 3.26.: *Mesh in the center of the structure with the material interface*

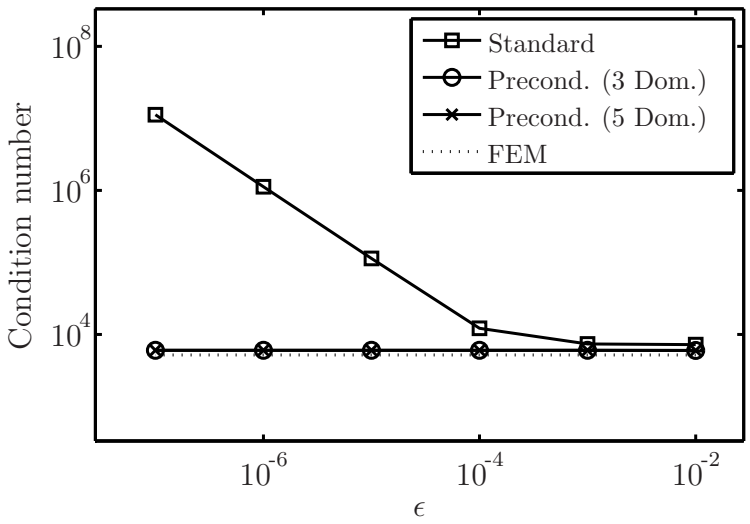


Figure 3.27.: *Condition number evaluated for different values of ϵ (material interface problem)*

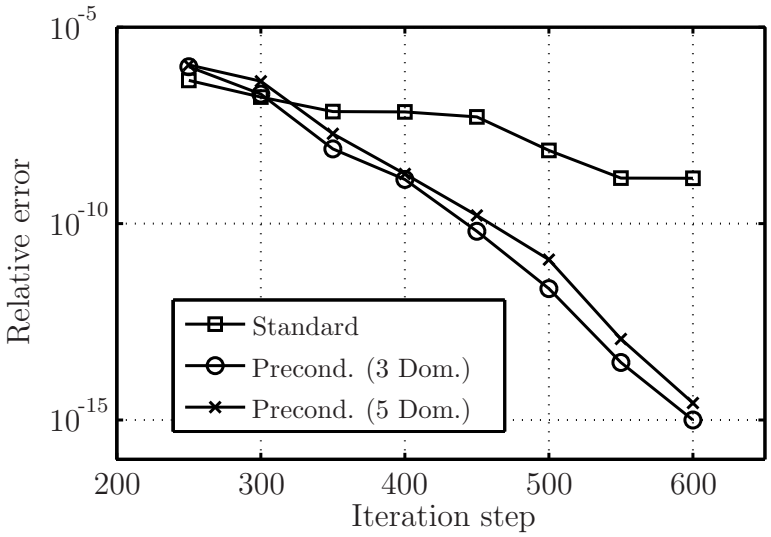


Figure 3.28.: *Relative error of the MINRES solver after different iteration steps (material interface problem, $\epsilon = 10^{-7}$)*

	Standard	Preconditioned (3 Dom.)	Preconditioned (5 Dom.)
Number of enriched DOFs	376	392	408
Number of DOFs $\boldsymbol{K}_{X,X}^{\Omega_i}$(avg.)	-	131	82
Number of DOFs $\boldsymbol{K}_{X,X}^{\Omega_i}$(max.)	-	136	88
Size of $\boldsymbol{Y}\boldsymbol{P}_X$	-	16x392	32x392

Table 3.8.: *Comparison of submatrices of the stiffness matrices for different versions of X-FEM (material interface problem, $\epsilon = 10^{-7}$)*

more enrichments have been split. Both domain decompositions result in domains which have approximately the same number of enriched degrees of freedom. Using the equations for the numerical effort discussed in section 3.3.2 we can estimate the computational effort in flops for each of the matrix decompositions. The result can be seen in table 3.9. Based on

	Preconditioned (3 Dom.)	Preconditioned (5 Dom.)
Flops LQ	$\approx$ 3e+5	$\approx$ 1.2e+6
Flops Chol. (avg.)	$\approx$ 7.4e+5	$\approx$ 1.7e+5
Total flops	$\approx$ 2.5e+6	$\approx$ 2e+6

Table 3.9.: *Estimates for the overall computational effort to calculate the preconditioner (material interface problem, $\epsilon = 10^{-7}$)*

these estimates we can guess that the decomposition into 5 subdomains is slightly more efficient. Please note that following this reasoning, both domain decompositions are more efficient than the naive approach of treating the whole structure as one large subdomain. The estimated number of flops would in that case be $\frac{376^3}{3} \approx 1.8e+7$.

Cracked structure

The second structure considered here is an isotropic material with the same material constants as material 1 in table 3.7. The structure contains a vertical crack with its tip at the center of the structure as shown in figure 3.29. Zero displacements are prescribed at the boundary on the left and a constant traction $\boldsymbol{t}$ on the right.

Again a simple mesh was used to discretize the structure consisting of 1458 equally sized elements. The mesh and different domain decompositions are shown in figure 3.30. The mesh and the enriched elements around the crack tip are shown in figure 3.31 and figure 3.32. Figure 3.31 shows the elements enriched by the crack tip enrichments and figure 3.32 the elements enriched by the Heaviside function. The choice of subdomains is independent of the enrichment radius, therefore, if the enrichment radius is small, some of the subdomains may contain no enrichments.

The condition number of the equation system was evaluated for different

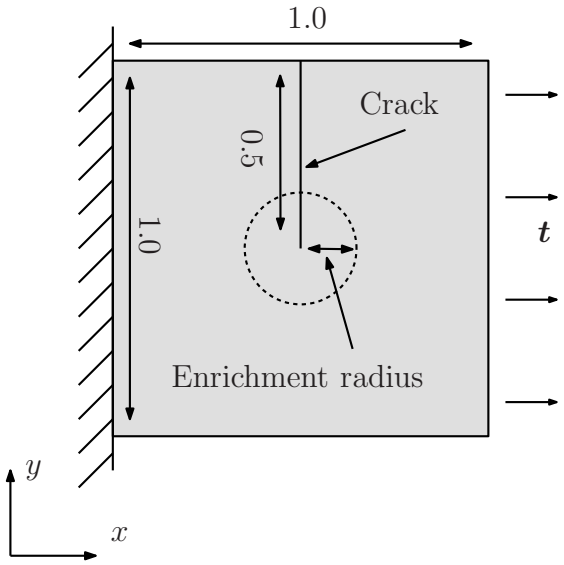


Figure 3.29.: *Structure with crack*

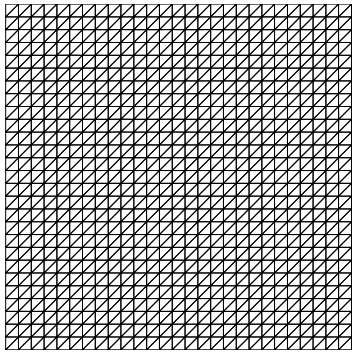
enrichment radii. The results are shown in figure 3.33. Again the condition number of the preconditioned version is only slightly higher than the condition number of the FEM equation system without any enrichment. No difference can be observed due to the different domain decompositions or enrichment radii. The equation systems of the standard version however become ill-conditioned for large enrichment radii.

The relative error of the MINRES solver depending on the number of iterations for an enrichment radius of 0.4 is shown in figure 3.34. No significant difference in the convergence behavior can be observed due to the different domain decompositions in the preconditioned version. But the difference between the preconditioned and the standard version is even more apparent than in the previous example.

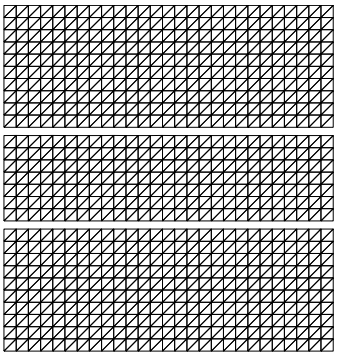
The properties of the submatrices of the stiffness matrices are compared in table 3.10. The effort for each Cholesky-decomposition decreases if more subdomains are used, but the computational effort for the LQ-decomposition increases.

The estimates for the total number of floating point operations are compared in table 3.11. Based on these estimates we can guess that the decomposition into 3 subdomains should be preferred if the enrichment radius is 0.4.

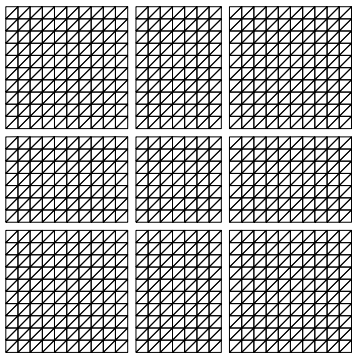
We can estimate the overall computational effort for treating the whole domain as one large subdomain to be $\frac{3686^3}{3} \approx 1.7e+10$.



(a) Mesh used to discretize the cracked structure



(b) Domain decomposed into 3 Subdomains



(c) Domain decomposed into 9 subdomains

Figure 3.30.: *Mesh used to discretize the cracked structure and different domain decompositions*

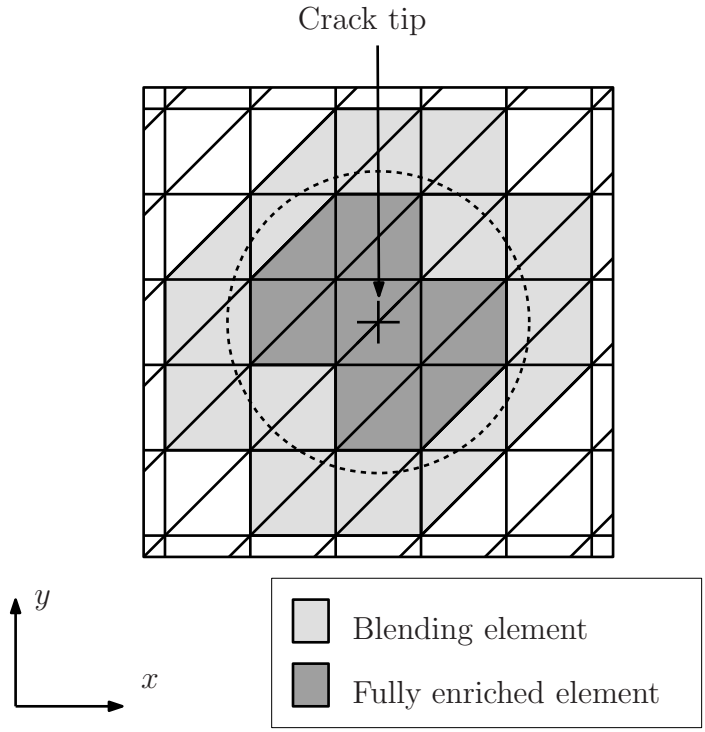


Figure 3.31.: *Elements enriched by the crack tip enrichment functions*

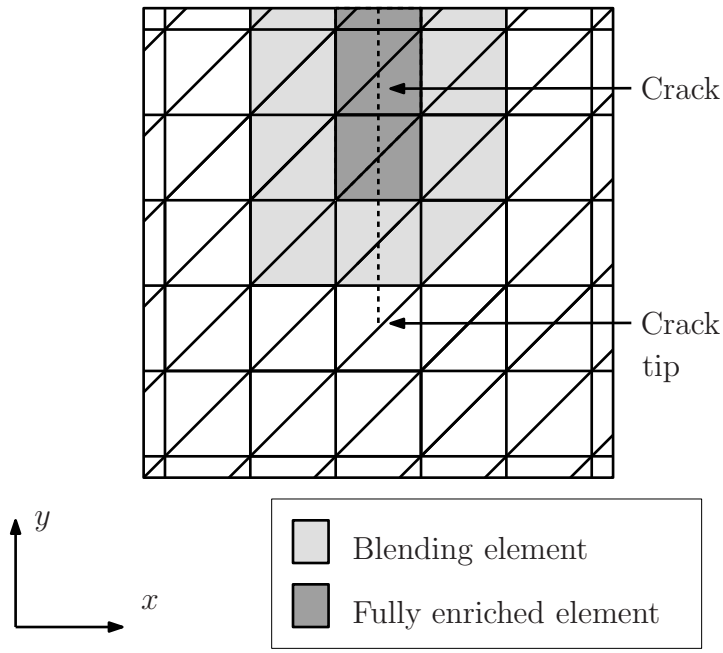


Figure 3.32.: *Elements enriched by the Heaviside-function*

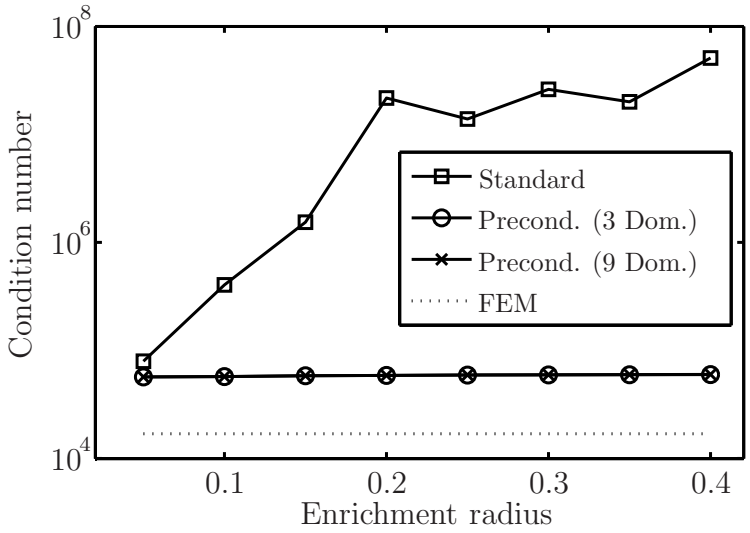


Figure 3.33.: *Condition number evaluated for different enrichment radii (cracked structure)*

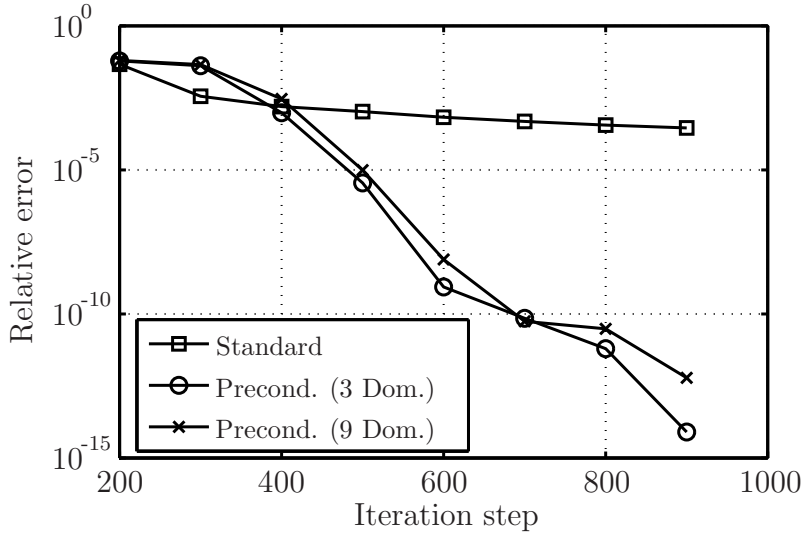


Figure 3.34.: *Relative error of the MINRES solver after different iteration steps (cracked structure, r_{max}=0.4)*

	Standard	Preconditioned (3 Dom.)	Preconditioned (9 Dom.)
Number of enriched DOFs	3686	4058	4458
Number of DOFs $\boldsymbol{K}_{X,X}^{\Omega_i}$(avg.)	-	1352	495
Number of DOFs $\boldsymbol{K}_{X,X}^{\Omega_i}$(max.)	-	1534	612
Size of $\boldsymbol{Y}\boldsymbol{P}_X$	-	268x4058	556x4458

Table 3.10.: *Comparison of submatrices of the stiffness matrices for different preconditioned versions of the X-FEM (cracked structure, r_{max}=0.4)*

	Preconditioned (3 Dom.)	Preconditioned (9 Dom.)
Flops LQ	$\approx$ 8.7e+8	$\approx$ 4.1e+9
Flops Chol. (avg.)	$\approx$ 8.2e+8	$\approx$ 4e+7
Total flops	$\approx$ 3.3e+9	$\approx$ 4.4e+9

Table 3.11.: *Estimates for the overall computational effort to calculate the preconditioner (cracked structure, r_{max}=0.4)*

Estimates of these kind are as mentioned in section 3.3.2 only a rule of thumb to estimate the real computational effort. But neglecting this, one may conclude that the domain decomposition really is a useful tool to maximize efficiency.
A big advantage is that these estimates can also be used to compare different domain decompositions before the procedure outlined here is actually applied.

Multiple cracks

The performance of the preconditioner is now addressed in a situation that is more likely to occur in everyday applications of X-FEM than the previous examples. In fracture mechanics there are often situations in which the interaction of multiple cracks is of particular interest. Therefore, the preconditioner is tested in such a situation.
The structure shown in figure 3.35 contains three vertical cracks originating from the upper and the lower boundary of the structure. Zero displacements

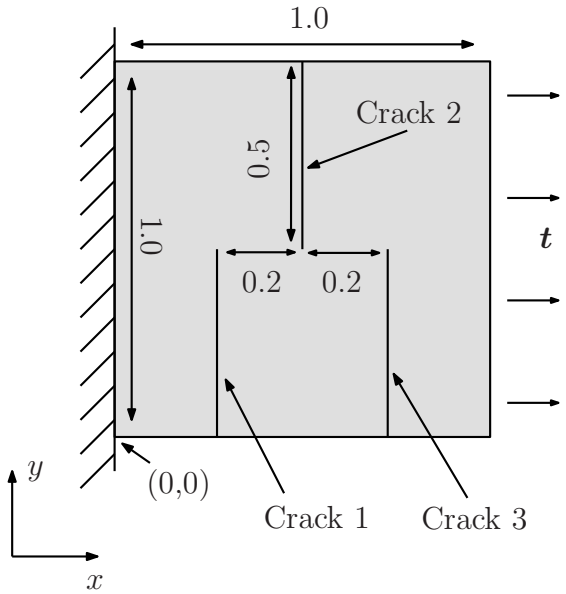
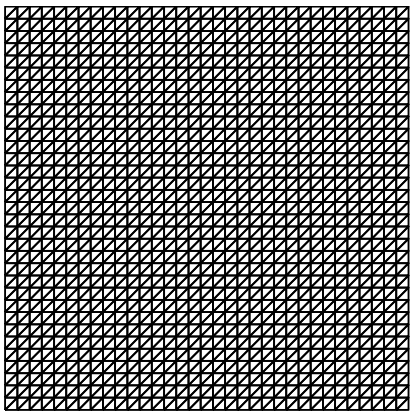


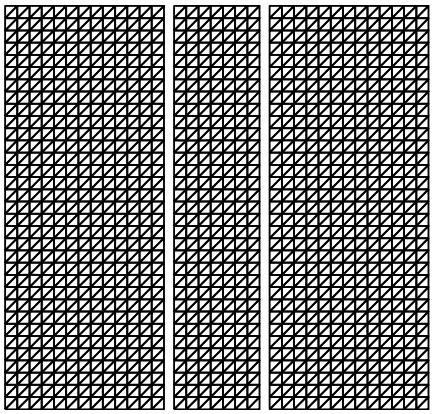
Figure 3.35.: *Structure containing multiple cracks*

are prescribed at $x = 0$ and a constant traction in positive x-direction is applied at $x = 1$. The values in table 3.7 associated with material 1 have been used to describe the elastic properties of the structure.
The structure was discretized using the mesh shown in figure 3.36(a) which

consists of 2178 triangular elements. A domain decomposition was per-



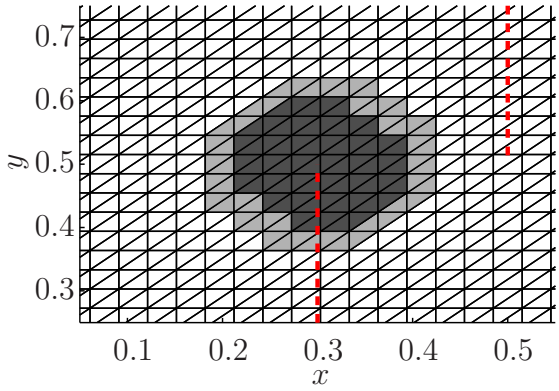
(a) Mesh



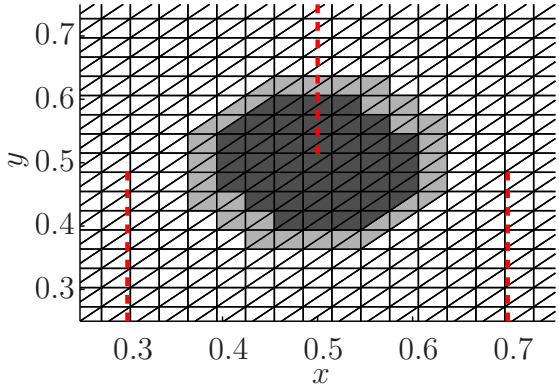
(b) Domain decomposition

Figure 3.36.: *Discretization of the structure containing multiple cracks*

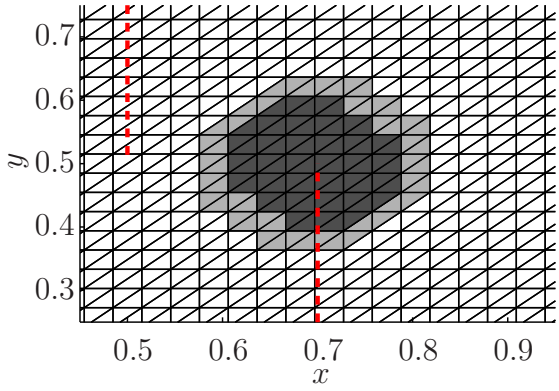
formed, the result is shown in figure 3.36(b). The enriched areas that have been used in combination with the crack tip enrichments are shown in figure 3.37(a)-3.37(c). They are chosen to overlap between the cracks in order to obtain a smooth representation of the stresses in the center of the structure. The areas in which the enrichments of different cracks overlap are shown in figure 3.38.



(a) Crack 1: Enriched area around the tip



(b) Crack 2: Enriched area around the tip



(c) Crack 3: Enriched area around the tip

Figure 3.37.: *Areas enriched by the crack-tip enrichments (blending elements are colored in light grey, cracks are indicated by dashed red lines)*

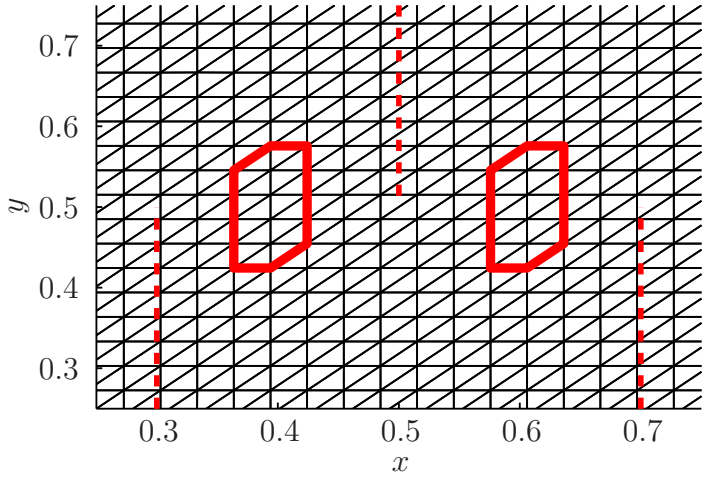


Figure 3.38.: *Overlap of the enriched areas (overlap indicated by solid red lines, cracks indicated by dashed red lines)*

The von Mises stress distribution resulting from an X-FEM calculation is depicted in figure 3.39. Although the overlap of the enriched areas allows for a smooth stress representation, it may cause ill-conditioning of the stiffness matrix. The underlying mechanisms, however, are much harder to identify since several enrichment functions associated with different cracks are involved.

The equation systems have been solved using the preconditioned and the standard version of X-FEM and computation times have been measured (Intel Core 2 Duo, 2 Ghz, 1GB Ram). The performance of a Matlab code is highly dependent on the way it is written and on its complexity. To make the measurements less dependent on a particular implementation of the preconditioner only the important substeps have been measured.

For the computation of the preconditioner these substeps are the Cholesky-decompositions, the LQ-decomposition and the evaluation of $\boldsymbol{Y}\boldsymbol{P}_X$. Each of the substeps has been repeated several times in a row and the total time was measured. The computation time for one particular substep was estimated to be the total time divided by the number of computations.

If function handles (a structure in Matlab) are passed to `minres()` the performance of the solver decreases. But since the matrix $\tilde{\boldsymbol{K}}$ is not explicitly available function handles have to be used. To be able to make a reasonable

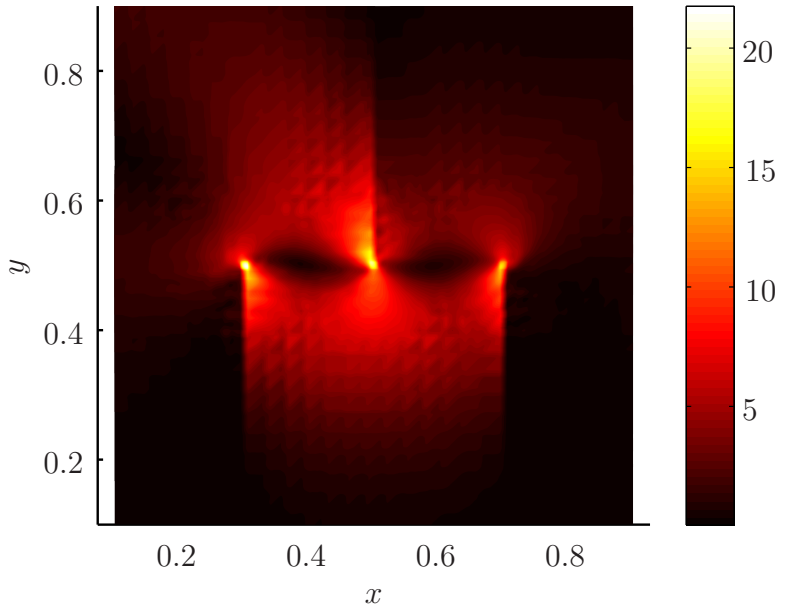


Figure 3.39.: *von Mises stresses for the structure containing multiple cracks*

comparison we therefore assumed that the main part of the computational effort in one iteration step of the MINRES solver consists of evaluating the matrix-vector product with $\boldsymbol{K}$ or $\tilde{\boldsymbol{K}}$ respectively. A time estimate for one matrix-vector product was again obtained by repeating the computation several times and dividing the total time by the number of computations. If the preconditioned version was used this involved solving several equation systems with triangular matrices due to the structure of $\boldsymbol{P}$. The time to solve the equations was then estimated to be the number of iterations the `minres()`-function uses to calculate an approximate solution with a relative error of 10^{-8} multiplied by the time estimate that was obtained for one calculation of the matrix-vector product with $\boldsymbol{K}$ or $\tilde{\boldsymbol{K}}$.

The results of the time measurements are shown in table 3.12. Due to the difference in matrix conditioning the solver needs much more iterations if the standard version of X-FEM is used. Although almost twice as much time is spent to calculate the matrix-vector product in the preconditioned version, the small number of iterations decreases the solution time significantly if compared to the standard version. As a result the total time

	Standard	Preconditioned
Condition number	1.0550e+007	4.6055e+005
Preconditioner calculation [s]	-	6.69
Matrix-vector product [s]	0.0018	0.0032
Iterations	11562	1041
Solution time [s]	20.81	3.24
Total time [s]	20.81	9.93

Table 3.12.: *Time measurements*

to obtain the result is significantly lower. The difference between the two methods would be even greater if different right-hand sides $\boldsymbol{f}$ are used, since the preconditioner has to be calculated only once.

3.4. Meshing

The grain structures in chapter 4 will be generated by a Voronoi tessellation. The result is a structure consisting of several convex grains. The Voronoi tessellation is especially useful for our purposes since it will make a random generation of grain structures possible. We therefore need an efficient strategy to generate meshes for these structures.

The meshing procedure will exploit the advantages of the X-FEM. Because the element edges do not have to align with grain boundaries and cracks in the X-FEM, the meshing procedure can be simplified significantly. This allows for a fully automated mesh generation which is still able to adapt to the complex geometries of a polycrystalline structure. The idea was first proposed by Menk and Bordas [35].

Duarte et al. [36] presented a method where adaptive meshing is performed around a crack tip in combination with Heaviside-enrichment. Bordas et al. [37] discussed an error estimator which could be used to control mesh adaptation in future applications. In most other applications of the X-FEM the mesh is not adapted.

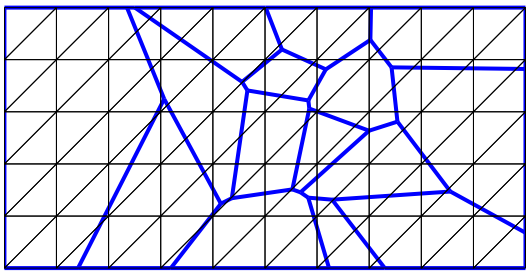
3.4.1. Refinement

Grain structures generated by a Voronoi tessellation contain grains of different sizes. Although the enrichment functions enable us to superimpose geometric features onto existing meshes, the mesh should be refined in those areas in which short grain boundaries or small grains are present. A simple meshing algorithm is described here which can be used in combination with the X-FEM. Due to its simple implementation the algorithm will always produce a mesh. It is therefore well suited for applications in which crack growth calculations for several grain structures should be performed in parallel without user intervention.

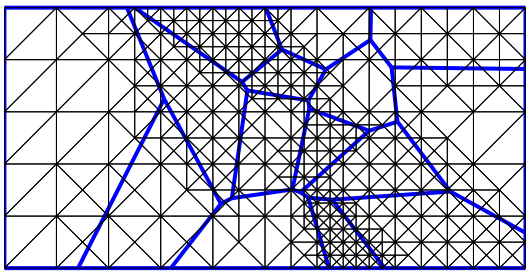
Suppose a grain structure is given and a background mesh as shown in figure 3.40(a) has been generated. The background mesh is now refined at certain locations. To refine a particular element, it is split along the line formed by the center of the longest element edge and the node opposite to this edge. This introduces two new elements. If the element splitting produces a hanging node in a neighboring element, the neighboring element is also refined by splitting the longest edge. This is done recursively to eliminate all hanging nodes. Please note, that by using this scheme we always obtain elements with a good aspect ratio.

The idea can be implemented efficiently if a refinement level is defined for each element. For equally sized elements as in figure 3.40(a) this level is initially set to 1 for each element. For two new elements obtained by element splitting, the refinement level is defined as the refinement level of the original element increased by 1. Thus, the coarsest elements in a set of elements are the ones with the smallest refinement level.

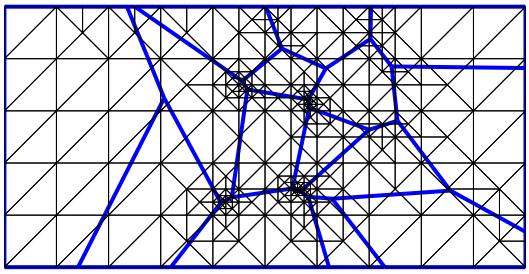
The code example 3.1 demonstrates the implementation using recursive function calls and a refinement level. The function `refineElement()` takes the element that should be refined as an input argument. `findElAttached()` determines the element which is attached to the longest element edge of `element`. This element is stored in `elAttached`. If the two elements have the same refinement level, splitting both elements will not introduce hanging nodes, since both elements are connected via their longest edge. In that case the first part of the `if`-statement is executed and the element is successfully refined. Otherwise the function `refineElement()` is called recursively for both elements again.



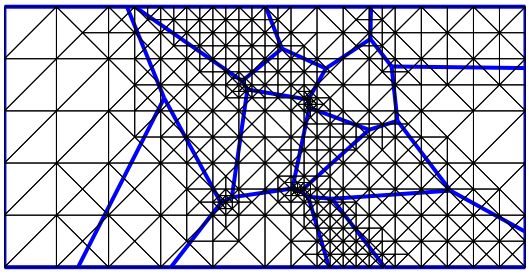
(a) Initial Mesh



(b) Mesh refined inside the grains



(c) Mesh refined along the boundaries



(d) Mesh refined inside the grains and around the boundaries

Figure 3.40.: *Mesh generation for a grain structure (grain boundaries are indicated by blue lines)*

Code example 3.1 Element refinement implemented using recursive function calls and the idea of a refinement level

```
refineElement(element)
   elAttached=findElAttached(element)
   if refinementLevel(element)=refinementLevel(elAttached)
      split(element)
      split(elAttached)
   else
      refineElement(elAttached)
      refineElement(element)
   end
return
```

For every grain, the coarsest element (i.e. the one with the smallest refinement level) is refined until a certain number of elements n_{gr} possesses nodes located inside that grain. The number n_{gr} should be dependent on the shape of the grain. We define the ratio:

$$r = \frac{\text{Area of smallest circle enclosing grain}}{\text{Area enclosed by the grain}} \tag{3.57}$$

The number n_{gr} is then chosen as:

$$n_{gr} = \hat{n}_{gr} r^2 \tag{3.58}$$

Setting $\hat{n}_g = 4$ and applying the procedure to the grain structure in 3.40(a) produces the mesh shown in figure 3.40(b). Clearly, the mesh is refined particularly inside the small grains and inside those grains that contain sharp edges. This strategy will be called grain refinement.
It is also important to refine the mesh along small grain boundaries. To refine the mesh along grain boundaries we refine the coarsest element that is cut by a certain boundary in the same way as previously explained. This refinement is done until a number n_b of elements is totally cut by a boundary. This will be called grain boundary refinement. Setting $n_b = 4$ produces the mesh shown in figure 3.40(c). Clearly the mesh is well adapted especially around the short boundaries. If both refinement strategies are performed consecutively, the mesh shown in figure 3.40(d) is obtained.
Since refinement should be done for the grains as well as for the grain

boundaries, we will employ the latter strategy (i.e. the combination of grain refinement and grain boundary refinement) for the numerical examples in chapter 4.

3.4.2. Element clustering

The remeshing scheme explained in section 3.4.1 can result in a high computational effort if a large number of elements and grains is involved. It is necessary to determine which elements are cut by which boundaries and which elements are inside a certain grain. These geometric operations are extremely time-consuming. The naive approach of testing each element and each boundary or grain for possible intersections is therefore not very efficient.
An efficient scheme is needed to determine which elements are in the vicinity of a certain boundary or grain. The mesh is therefore subdivided into several element clusters. Clustering is performed at different levels. At each level the element clusters of the previous level are split. At level 1 the mesh is split into four (possibly empty) element clusters. Using the maximum

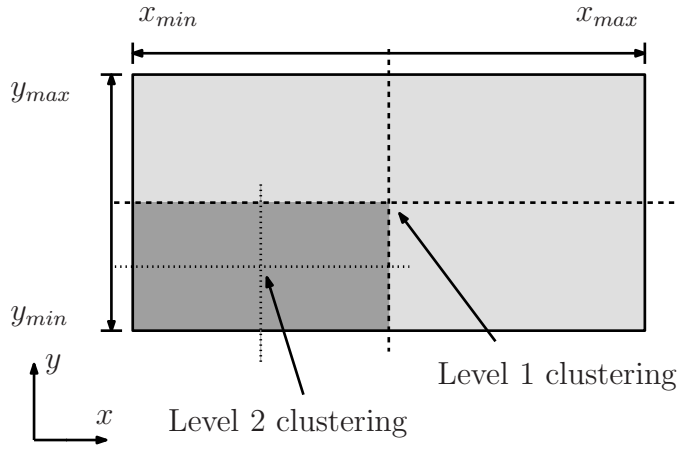


Figure 3.41.: *Visualization of the clustering scheme*

and minimum values for the x and y-coordinates of th structure one can

determine center coordinates by:

$$x_c := \frac{x_{min} - x_{max}}{2} \tag{3.59}$$

$$y_c := \frac{y_{min} - y_{max}}{2} \tag{3.60}$$

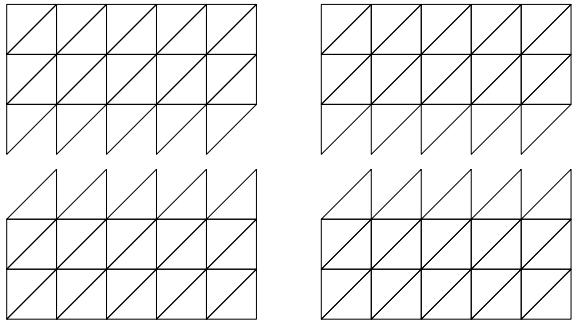
$$\tag{3.61}$$

Using these center coordinates one can split the structural domain in an upper left, an upper right, a lower left and a lower right area. The upper right area for instance would be formed by all points with $x > x_c$ and $y > y_c$. The mesh is then subdivided into four element clusters. All elements whose center is located in the same area form a cluster. This is called level 1 clustering.

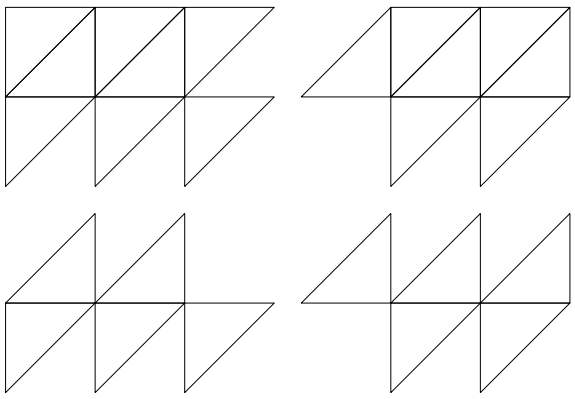
Each element cluster is divided into four smaller clusters. This is done by treating the cluster as a new mesh and applying the clustering idea again. We will call this level 2 clustering.

The idea is visualized in figure 3.41. Four areas obtained from level 1 clustering cover the structural domain. Each of these areas can again be split into four areas as shown for the lower left area. Performing element clustering for the initial mesh in figure 3.40(a), we obtain the first level element clusters as shown in figure 3.42(a). Performing element clustering at the second level, we obtain for the upper right cluster in figure 3.42(a) four new clusters, as shown in figure 3.42(b). This way a tree structure is generated. In this example we refine ourselves to two levels, although the scheme can be extended to clustering on more than two levels.

To determine which elements intersect with a grain boundary or a grain, one can use this tree structure very efficiently: For each element cluster we can determine a bounding box. The bounding box of a cluster is a rectangle which contains all elements inside that cluster. It can be determined by finding the minimal and the maximal values of the x- and the y-coordinates of all element nodes inside the cluster. Starting with all the level 1 element clusters, one detects possible intersections of elements with grains or grain boundaries by determining intersections with this bounding box. Only if possible intersections are detected for a certain element cluster, the clusters obtained by splitting this particular cluster on level 2 are checked for possible intersections. This is performed recursively at all levels. If the lowest



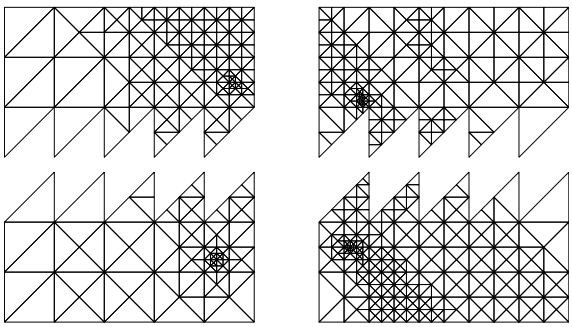
(a) Initial Mesh split into four element clusters



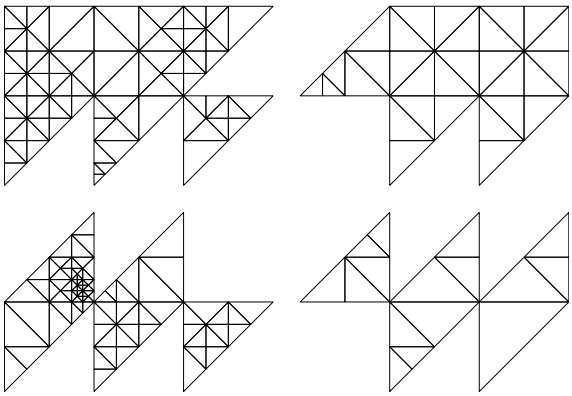
(b) Element cluster of the initial mesh in the upper right split into four smaller clusters

Figure 3.42.: *Subdivision of the initial mesh in a cluster structure*

level is reached, all elements that are forming the clusters for which possible intersections were detected are considered. The number of elements that possibly intersect with a grain or a grain boundary is at the lowest level significantly smaller than the total number of elements.
If elements are refined, the tree structure has to be updated. This can simply be done by replacing each refined element by the two new elements in each cluster in which it appears. The resulting clustering for grain re-



(a) Final Mesh split into four element clusters



(b) Element cluster in the upper right of the final mesh split into four smaller clusters

Figure 3.43.: *Cluster structure after mesh refinement*

finement and grain boundary refinement for the mesh in figure 3.40(a) is visualized in figures 3.43(a) and 3.43(b).
Subsequent element refinements for the same boundary or grain can be done

	Naive	Clustering
Grain boundary refinement	13069	3574
Grain refinement	6164	1600
Grain and grain boundary refinement	40808	8077

Table 3.13.: *Number of geometric operations needed to perform mesh refinement with and without element clustering*

by considering all the previously determined elements and the refined elements in addition. Therefore, for each boundary or grain, the tree structure has to be used only once.
To address how much of the costly geometric operations can be avoided by the clustering approach, we consider the refinements performed in section 3.4.1 in figure 3.40(b), 3.40(c) and 3.40(d). We compare the naive approach of checking every element with possible grain or grain boundary intersections with the element clustering approach on two levels.
In the clustering approach a geometric operation is either the test for the intersection of a grain boundary or a grain with an element, or the test for a possible intersection with the bounding box of an element cluster.
The results are compared in table 3.13. Obviously the number of geometric operations can be significantly reduced by using the clustering approach.
Generally, it was found that for the more complex examples in this thesis a clustering approach on three levels was an efficient alternative.

3.5. Contact Modeling

When applying the algorithmic procedure developed here so far to solder joints we are mainly interested in the deformation behavior under cyclic loading. If cracks are present in the structure it is likely that the crack faces overlap in the simulation. To prevent this, we have to add constraints to our problem.
In this thesis the constrained problem is solved by using a penalty method. More sophisticated approaches for dealing with contact problems in combination with the X-FEM were discussed by Ribeaucourt et al. [38] and Dolbow et al. [39].

Due to the positive definiteness of $\boldsymbol{K}$ the numerical solution of the structural problem can be interpreted as finding a vector $\boldsymbol{a}$ such that the elastic energy of the system is minimized:

$$\min |\frac{1}{2}\boldsymbol{a}^T\boldsymbol{K}\boldsymbol{a} - \boldsymbol{a}^T\boldsymbol{f}| \tag{3.62}$$

We assume that the constraints are given by a sufficiently smooth function η which is characterized by:

$$\eta(\boldsymbol{a}) \begin{cases} = 0 \text{ if the crack faces do not overlap} \\ > 0 \text{ if the crack faces overlap} \end{cases} \tag{3.63}$$

We want to minimize (3.62) subject to the condition $\eta(\boldsymbol{a}) = 0$. We therefore add an additional term to the minimization problem with $\epsilon > 0$:

$$\min |\frac{1}{2}\boldsymbol{a}^T\boldsymbol{K}\boldsymbol{a} + \frac{1}{\epsilon}\eta(\boldsymbol{a}) - \boldsymbol{a}^T\boldsymbol{f}| \tag{3.64}$$

For $\epsilon \to 0$ the minimum of (3.64) converges to the minimum of (3.62) subject to $\eta(\boldsymbol{a}) = 0$. The idea of the penalty method is to use a small positive value for ϵ.

For non-zero values of ϵ and a sufficiently smooth function η, the expression (3.64) can be minimized using a Newton method. We define:

$$\nabla\eta(\boldsymbol{a}) := \begin{bmatrix} \frac{\partial\eta(\boldsymbol{a})}{\partial a_1} \\ \vdots \\ \frac{\partial\eta(\boldsymbol{a})}{\partial a_n} \end{bmatrix} \tag{3.65}$$

A local minimum of (3.64) is found if the gradient vanishes, that is:

$$\boldsymbol{K}\boldsymbol{a} + \frac{1}{\epsilon}\nabla\eta(\boldsymbol{a}) = \boldsymbol{f} \tag{3.66}$$

Linearizing (3.66) around $\boldsymbol{a}$, we obtain:

$$\boldsymbol{K}\left[\boldsymbol{a}+\Delta\boldsymbol{a}\right]+\frac{1}{\epsilon}\nabla\eta(\boldsymbol{a})+\boldsymbol{K}_P(\boldsymbol{a})\Delta\boldsymbol{a}=\boldsymbol{f} \tag{3.67}$$

$$\boldsymbol{K}_P(\boldsymbol{a}):=\frac{1}{\epsilon}\begin{bmatrix}\frac{\partial^2\eta(\boldsymbol{a})}{\partial x_1\partial x_1}\cdots\frac{\partial^2\eta(\boldsymbol{a})}{\partial x_1\partial x_n}\\ \vdots\ddots\vdots\\ \frac{\partial^2\eta(\boldsymbol{a})}{\partial x_n\partial x_1}\cdots\frac{\partial^2\eta(\boldsymbol{a})}{\partial x_n\partial x_n}\end{bmatrix} \tag{3.68}$$

We can easily put this in the form of an iterative method where $\boldsymbol{a}_i$ denotes the i-th iterate:

$$\Delta\boldsymbol{a}=\left[\boldsymbol{K}+\boldsymbol{K}_P(\boldsymbol{a}_i)\right]^{-1}\left[\boldsymbol{f}-\frac{1}{\epsilon}\nabla\eta(\boldsymbol{a}_i)-\boldsymbol{K}\boldsymbol{a}_i\right] \tag{3.69}$$

$$\boldsymbol{a}_{i+1}:=\boldsymbol{a}_i+\Delta\boldsymbol{a} \tag{3.70}$$

Thus, in each iteration step one has to calculate $\boldsymbol{K}_P$ and solve a linear equation system with $\boldsymbol{K}+\boldsymbol{K}_P$.
An approximation η^h of η can be constructed as follows. Assume that an element is completely cut by a crack as shown in figure 3.44. $\boldsymbol{n}$ denotes the

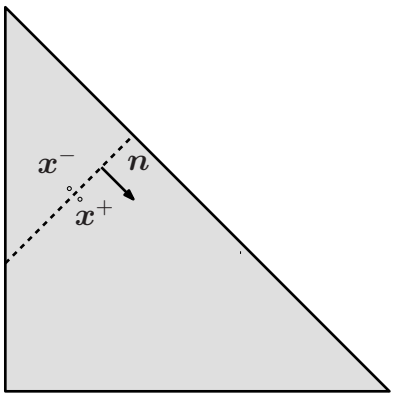


Figure 3.44.: *Element cut by a crack*

unit vector perpendicular to the crack. In the center of the crack segment we can obtain the points $\boldsymbol{x}^-$ and $\boldsymbol{x}^+$ by making an infinitesimal step in the positive and the negative direction of $\boldsymbol{n}$. Crack faces inside the element overlap if:

$$(\boldsymbol{u}(\boldsymbol{x}^+)-\boldsymbol{u}(\boldsymbol{x}^-))^T\boldsymbol{n}>0 \tag{3.71}$$

This is however just a sufficient but not a necessary condition. To obtain a necessary condition for small deformations (3.71) has to be evaluated for all points along the crack. However, the evaluation in the center of the crack segment in an element can be interpreted as a numerical approximation to a necessary condition.
It is easy to construct matrices $\boldsymbol{N}^+$ and $\boldsymbol{N}^-$ such that:

$$\boldsymbol{u}(\boldsymbol{x}^+) = \boldsymbol{N}^+\boldsymbol{a} \tag{3.72}$$

$$\boldsymbol{u}(\boldsymbol{x}^-) = \boldsymbol{N}^-\boldsymbol{a} \tag{3.73}$$

The entries of these matrices are obtained by evaluating the enrichment functions at $\boldsymbol{x}^+$ and $\boldsymbol{x}^-$. Since cracks are introduced only by enrichment functions it suffices to evaluate the enrichment functions of the corresponding element. For the j-th element cut by a crack we can construct:

$$\boldsymbol{N}_P^j = \boldsymbol{N}^+ - \boldsymbol{N}^- \tag{3.74}$$

$$\eta_j = \begin{cases} 1 \text{ if } \boldsymbol{n}^T\boldsymbol{N}_P^j\boldsymbol{a} > 0 \\ 0 \text{ otherwise} \end{cases} \tag{3.75}$$

The matrix $\boldsymbol{N}_P$ is constructed by:

$$\boldsymbol{N}_P = \begin{bmatrix} \sqrt{\eta_1}\boldsymbol{N}_P^1 \\ \vdots \\ \sqrt{\eta_n}\boldsymbol{N}_P^n \end{bmatrix} \tag{3.76}$$

Depending on whether the crack faces in a certain element overlap, several of the lines in $\boldsymbol{N}_P$ maybe zero.
We obtain $\boldsymbol{K}_P^h$ (which is the numerical approximation to $\boldsymbol{K}_P$) by:

$$\boldsymbol{K}_P^h = \boldsymbol{N}_P^T\boldsymbol{N}_P \tag{3.77}$$

Although the function $\frac{1}{\epsilon}\eta^h$ (the approximation to the penalty term in (3.64)) is not actually calculated, its explicit form would be:

$$\frac{1}{\epsilon}\eta^h = \boldsymbol{a}^T\boldsymbol{K}_P^h(\boldsymbol{a})\boldsymbol{a} \tag{3.78}$$

The solution of $\boldsymbol{K} + \boldsymbol{K}_P^h$ for varying $\boldsymbol{K}_P^h$ can be coupled efficiently with the preconditioner introduced in section 3.3. The preconditioner enables us to solve equation systems with $\boldsymbol{K}$, since the preconditioned version of $\boldsymbol{K}$ is well-conditioned. But for the Newton method we have to solve equation systems with $\boldsymbol{K} + \boldsymbol{K}_P^h$. Applying the same preconditioner may not result in well-conditioned equation systems. Calculating a new preconditioner for $\boldsymbol{K} + \boldsymbol{K}_P^h$ at each time step of a transient calculation and at each subsequent step of the Newton method would not be efficient.
But since the rank of the matrix $\boldsymbol{K}_P^h$ is usually much lower than the rank of the matrix $\boldsymbol{K}$, we can solve an equation system with $\boldsymbol{K}$ instead of $\boldsymbol{K} + \boldsymbol{K}_P^h$ and update the solution afterwards. The Woodbury matrix identity [34] can be formulated for our problem as (cf. Appendix B):

$$\left[\boldsymbol{K} + \boldsymbol{N}_P^T \boldsymbol{N}_P\right]^{-1} = \boldsymbol{K}^{-1} - \boldsymbol{K}^{-1} \boldsymbol{N}_P^T \left[\boldsymbol{I} + \boldsymbol{N}_P \boldsymbol{K}^{-1} \boldsymbol{N}_P^T\right]^{-1} \boldsymbol{N}_P \boldsymbol{K}^{-1} \quad (3.79)$$

Assume that we already solved $\boldsymbol{K}\hat{\boldsymbol{a}} = \boldsymbol{f}$. By multiplying equation (3.79) from the right with $\boldsymbol{f}$ we obtain:

$$\boldsymbol{a} = \hat{\boldsymbol{a}} - \boldsymbol{K}^{-1} \boldsymbol{N}_P^T \left[\boldsymbol{I} + \boldsymbol{N}_P \boldsymbol{K}^{-1} \boldsymbol{N}_P^T\right]^{-1} \boldsymbol{N}_P \hat{\boldsymbol{a}} \quad (3.80)$$

$\boldsymbol{a}$ is the solution of $\left[\boldsymbol{K} + \boldsymbol{K}_P^h\right] \boldsymbol{a} = \boldsymbol{f}$. The factor $\boldsymbol{K}^{-1} \boldsymbol{N}_P^T$ appears twice. If this factor is known in advance, calculating $\boldsymbol{a}$ with (3.80) reduces to solving an equation system with $\boldsymbol{K}$, solving a small equation system with $\left[\boldsymbol{I} + \boldsymbol{N}_P \boldsymbol{K}^{-1} \boldsymbol{N}_P^T\right]$, evaluating a small number of matrix vector products and a vector addition.
There is a simple way to obtain $\boldsymbol{K}^{-1} \boldsymbol{N}_P^T$. We can calculate the matrix $\boldsymbol{K}^{-1} \hat{\boldsymbol{N}}_P^T$ with $\hat{\boldsymbol{N}}_P^T$ given by:

$$\hat{\boldsymbol{N}}_P = \begin{bmatrix} \frac{1}{\sqrt{\epsilon}} \boldsymbol{N}_P^1 \\ \vdots \\ \frac{1}{\sqrt{\epsilon}} \boldsymbol{N}_P^n \end{bmatrix} \quad (3.81)$$

Only a small number of equation systems have to be solved additionally since the rank of $\hat{\boldsymbol{N}}_P^T$ is determined by the number of cracked elements. From this we can obtain $\boldsymbol{K}^{-1} \boldsymbol{N}_P^T$ by setting those rows to zero which cor-

respond to elements in which η_j is zero.
Hence, the calculation of $\boldsymbol{K}^{-1}\hat{\boldsymbol{N}}_P^T$ has to be performed only once before the application of the Newton method. In the case of a transient calculation with a fixed crack geometry, it even suffices to calculate the matrix once before the whole calculation.
If calculations have been performed for a previous time step (cf. section 4.4.2), then the corresponding displacements have been used as a starting value for the Newton method in the current time step. A tolerance of 10^{-4} was found to be a good compromise between accuracy and the number of iterations.

3.6. Transient simulations and creep with the X-FEM

In this section it is explained how the methods described so far can be applied to transient problems involving plastic material behavior described by a creep law. An explicit solution procedure is used based on the work of Zienkiewicz and Cormeau [40]. The main challenge in using the concepts explained so far for creep problems is to eliminate numerical difficulties which occur due to the use of the weakly singular enrichments.
Another application of the X-FEM to transient problems was described by Menouillard et al. [41]. A dynamic linear elastic problem was solved using the Newmark method combined with the X-FEM. Due to the use of a lumped mass matrix they were able to calculate the solution without solving equation systems. The method can also be classified as an explicit approach.
For plastic-elastic fracture mechanics appropriate enrichment functions to describe the weakly singular fields in front of the crack tip were discussed by Elguej et al. [42]. Time-independent plasticity for isotropic materials was considered.
Liang et al. [43] used the X-FEM in combination with a creep law before. They used an implicit solver to solve their equation systems. The numerical difficulties discussed in this section were not discussed in their paper, it is possible that they did not appear because of the implicit solution process.

An explicit approach as it is used here, is simpler to implement than an implicit one. Furthermore, convergence problems, which may appear if implicit solvers are used, cannot occur if explicit solvers are used.
It must be mentioned that, although the numerically determined enrichment functions (cf. section 3.2) are able to represent arbitrary strain singularities for two-dimensional linear elastic problems appropriately, this may not be true any more if a creep law is used additionally to describe the transient behavior. The order of the strain singularity and its shape may change in that case.
However, the numerically determined enrichment functions will still be useful to introduce crack tips and re-entrant corners onto meshes that do not conform to the geometry. Thus, the main advantage of using them in chapter 4 is that they will allow us to realize the automatic meshing as described in section 3.4 procedure for solder joints containing cracks.
We are interested in performing thermomechanical transient simulations. Assume that a temperature profile $T(t)$ together with appropriate displacement boundary conditions are given for a structure. Using (2.17) and the strain representation in equation (2.2) we obtain the weak formulation at time t as:

$$\text{Find } \boldsymbol{u}(t) \in \left(H^1_{\Gamma_D}(\Omega)\right)^2 \text{ such that for all } \boldsymbol{v} \in \left(H^1_{\Gamma_D}(\Omega)\right)^2\text{:}$$

$$\int_\Omega \boldsymbol{\epsilon}(\boldsymbol{v}) : \boldsymbol{D} : \boldsymbol{\epsilon}(\boldsymbol{u}(t))da = \int_\Omega \boldsymbol{\epsilon}(\boldsymbol{v}) : \boldsymbol{D} : (\boldsymbol{\epsilon}^{th}(t) + \boldsymbol{\epsilon}^{cr}(t))da \qquad (3.82)$$

If the thermal strains and the creep strains on the right hand side are known we can solve the problem numerically with the previously described techniques.
To perform a transient calculation the time axis must be discretized. We do this by choosing a fixed time step Δt. Assume that a calculation at time t has already been performed. The creep and the thermal strains at time $t+\Delta t$ are needed to evaluate the right hand side of (3.82) for the subsequent time step. The thermal strains are obtained by evaluating the temperature profile T at $t + \Delta t$ and using equation (2.12). An approximation for the creep strains is given by:

$$\boldsymbol{\epsilon}^{cr}(t + \Delta t) \approx \boldsymbol{\epsilon}^{cr}(t) + \frac{\partial \boldsymbol{\epsilon}^{cr}}{\partial t}(\boldsymbol{\sigma}(t), T, t)\Delta t \qquad (3.83)$$

This approach was proposed by Zienkiewicz and Cormeau [40] for viscoelastic materials. The stresses in (3.83) are given by:

$$\boldsymbol{\sigma}(t) = \boldsymbol{D} : (\boldsymbol{\epsilon}(t) - \boldsymbol{\epsilon}^{th}(t) - \boldsymbol{\epsilon}^{cr}(t)) \tag{3.84}$$

A numerical approximation to $\boldsymbol{\epsilon}(t)$ can be calculated using the displacement vector $\boldsymbol{a}(t)$ describing the numerical solution at time t.
If the time step is too large the solution becomes unstable. Zienkiewicz and Cormeau [40] proposed time steps depending on the maximum value of $\frac{\partial \boldsymbol{\epsilon}^{cr}}{\partial t}(\boldsymbol{\sigma}(t), T, t)$ in the structural domain. But in combination with the X-FEM this approach is problematic. Because the numerical solution is able to display infinite stresses and strains, the creep rates may theoretically also become infinite. Therefore, we would have to choose an infinitely small time step to compensate this.
In practice the creep rates are only evaluated at the Gauss points, but still, the maximum creep rate would be dependent on the location of the Gauss point. Therefore, the time step and the progress of the algorithm would be dependent on the location of the Gauss points and thus on the order of the Gaussian quadrature. Since the enriched elements are usually integrated with a high order a very small time step may be necessary to guarantee a stable algorithm.
Equation (3.83) is only an approximation to the true value of the creep strain at time $t + \Delta t$ given by:

$$\boldsymbol{\epsilon}^{cr}(t + \Delta t) = \boldsymbol{\epsilon}_{cr}(t) + \int_t^{t+\Delta t} \frac{\partial \boldsymbol{\epsilon}_{cr}}{\partial \tau}(\boldsymbol{\sigma}(\tau), T, \tau) d\tau \tag{3.85}$$

The approximation in equation (3.83) can be motivated as follows: Assume that the total strains $\boldsymbol{\epsilon}(t)$, the thermal strains $\boldsymbol{\epsilon}^{th}(t)$, the temperature $T(t)$ and the creep strains $\boldsymbol{\epsilon}^{cr}(t)$ are known. If all these quantities would remain unchanged for all times between t and $t+\Delta t$, then equation (3.85) simplifies to equation (3.83).
We will use a simple approach to guarantee the stability of the solution for all time steps. From all the assumptions mentioned above, only the assumption of constant total strains $\boldsymbol{\epsilon}(t)$ for all times between t and $t + \Delta t$ is needed. This assumption enables us to evaluate the integral in (3.85) numerically

using a step size $\Delta\tau << \Delta t$, but without considering the global behavior of the structure (i.e. without solving another equation system).
To explain this more precisely we consider the code example 3.2. The code

Code example 3.2 Stabilized evaluation of the creep strain

```
deltaTau=deltaT/n
for i=1:n
    Tmp=temp(t)
    sigma=stress(Gpt,a,epsCr,Tmp,t)
    epsCr=crStrain(epsCr,Tmp,sigma,deltaTau)
    t:=t+deltaTau
end for
```

demonstrates the numerical integration of the creep strain for one Gauss point `Gpt`. The displacements obtained from a solution at time `t` are stored in `a`. The time step Δt is stored in `deltaT`, the small time step $\Delta\tau := \frac{\Delta t}{n}$ is stored in `deltaTau`, n is some integer stored in `n`. The function `temp()` evaluates the temperature at time `t`. The numerical integration of (3.85) is performed by the `for`-loop. The function `stress()` is an evaluation of (3.84). The function `crStrain()` is an evaluation of equation (3.83) but the small time step $\Delta\tau$ is used instead of Δt.
The effect on the stability of the simulation can be interpreted as follows: If the creep rates are extremely high at some point, evaluation of (3.83) may lead to unreasonably high creep strains, because it is only an approximation of (3.85). The implementation discussed here allows for a relaxation of the stresses within a time step Δt. Thus, the creep rate also decreases within Δt. Hence, at time $t + \Delta t$ the creep strain is generally lower than the one obtained by evaluating (3.83).
In the numerical experiments conducted in chapter 4 this approach is used with a value $\Delta\tau = \frac{\Delta t}{100}$ for all elements for which the value $f(\sigma_{vm}, T)\Delta t$ exceeds 0.1 at one or more Gauss points. $f(\sigma_{vm}, T)$ is the function used to define the creep strain. For all other elements the approach in equation (3.83) is used.
No statement was made about how the choice of $\Delta\tau$ and Δt is related to the numerical error. But it is obvious that for a given choice of Δt, the approach discussed here is an improvement of the approach proposed by Zienkiewicz and Cormeau [40].

4. Application of X-FEM to solder joints in electronic devices

In this chapter the methods previously explained are applied to the calculation of crack growth in solder joints during thermal cycling. This is the first application of X-FEM to this problem.
Section 4.1 gives a short introduction to solder joints in electronic devices, the soldering process and thermomechanical fatigue. The methods currently employed to predict the lifetime of solder joints are discussed in section 4.2. A new methodology is motivated and introduced in 4.3. This includes a discussion about the influence of microstructural features on the damage process. The discussion is based on the literature that is currently available on that subject. The new methodology enables crack growth calculations in solder joints based on microstructural features of the joint. It employs the random generation of microstructures which is explained in section 4.3.2. Constitutive equations describing the mechanical properties of the joint microstructure are determined in section 4.4.1. Crack growth simulations in solder joints have been performed and the results are presented in section 4.4.2.
The idea for the solder joint model, the grain structure generation and the crack growth criterion was first presented by Menk and Bordas [35].

4.1. Solder joints in electronic devices

In order to realize electronic devices (e.g. control units for cars) different components such as resistors, chips etc. are needed. These components are connected such that the desired functionality of the final device is achieved. The connections are formed by soldering the components on a circuit board with copper wirings on its surface. Therefore, the solder joints form a mechanical as well as an electrical connection. The mechanical connection ensures the attachment of the components to the board while the electrical connection enables the functionality of the device.
Before the actual soldering, the solder paste is printed onto those locations on the board to which the components should be attached. The components are placed on top of the paste. During soldering the surrounding temperature is increased such that the metal components of the solder paste liquefy completely. Afterwards the surrounding temperature is decreased and the solder solidifies to form the joints.
The solder paste contains different powdered metals mixed with a flux. The

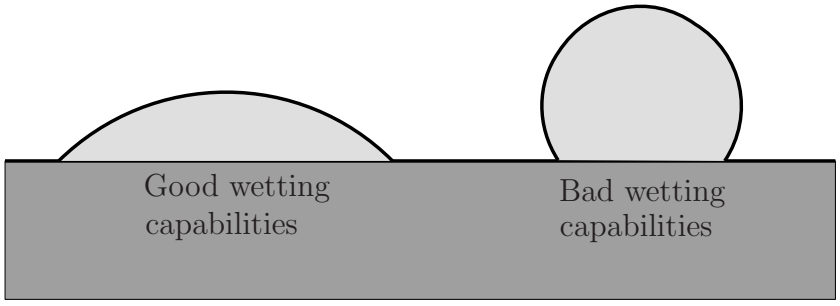


Figure 4.1.: *Wetting capabilities of the solder*

flux is a chemical which removes oxidations from the metal contacts that are to be joined. The contacts are often made from copper. At elevated temperatures oxidations form easily at the copper surface. Oxidations impair the wetting capabilities of the liquid solder. This is visualized in figure 4.1. The wetting capability is good if the contact area between the solder and the copper is big compared to the solder volume. If the wetting properties of the solder are poor, the final joint will be of a poor quality since the contact area between the component and the board is narrow. The conductivity may therefore be impaired and the connection is likely to fail

if mechanical loads are applied. The flux vaporizes completely during the soldering process.

The composition of the different metals forming the powder in the solder paste is chosen such that the alloy has certain manufacturing and reliability properties. An important property is the liquidus temperature. The liquidus temperature is the lowest temperature at which the solder is completely liquid. To form a good connection by soldering, the solder paste should be completely liquid at some point of the soldering process. If the liquidus temperature is too high, the components may be damaged in the soldering process. If it is too low, the solder will be less resistant to mechanical loads if the device is used in a high temperature environment such as the vicinity of a cars engine.

For a long time the SnPb solder alloy has been successfully used in the electronics industry. It consists of roughly 60 wt% tin and 40 wt% lead. Since

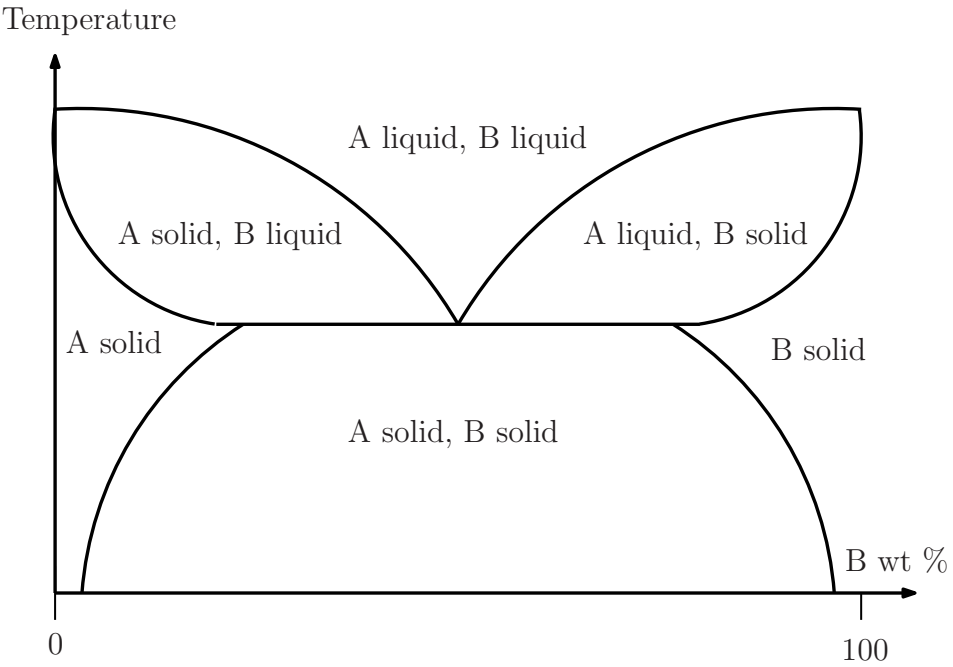


Figure 4.2.: *Example of a phase diagram for a two-component alloy*

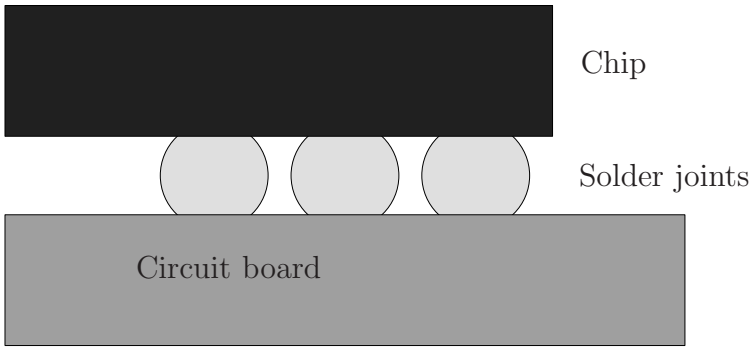
July 2006 lead-containing electronics are banned in the European union due to environmental concerns and health issues. The new solder alloy is the SnAgCu alloy which consists of roughly 95-96 wt% tin, 3-4 wt% silver and a small percentage of copper (0.5-0.8 wt%). Both alloys are known as near-eutectic alloys.

Near-eutectic alloys have good manufacturing properties. To motivate this we imagine an alloy formed by two different metals. A phase diagram for this alloy could look like the one in figure 4.2. In fact, this phase diagram has all the characteristics of the SnPb phase diagram. Phase diagrams describe the aggregate state of the alloy components at different temperatures. If for some alloy composition and temperature a component is solid, it may contain small amounts of the other component. Thus, "A solid" in figure 4.2 should be interpreted as a crystal formed by component "A" with atoms of "B" dissolved in it.

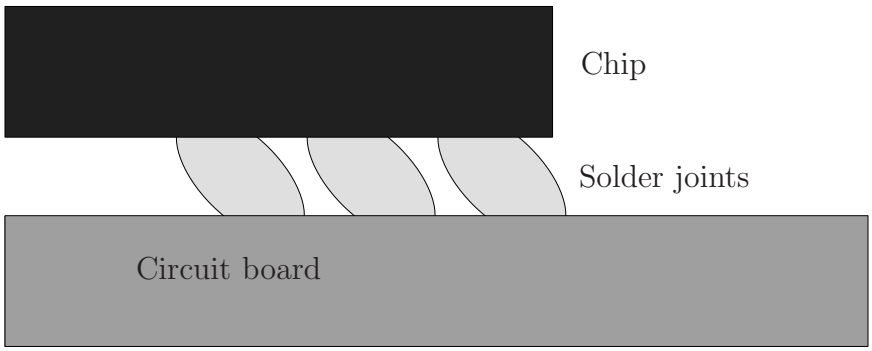
From figure 4.2 we can see that there is one particular composition which has a very low liquidus temperature. At this temperature all other alloy compositions contain solid components, i.e. their liquidus temperature is higher. This particular alloy composition is known as the eutectic composition. A direct transition from the solid state to a liquid state is possible. For three-component alloys, such as SnAgCu, phase diagrams also exist, but they cannot be displayed that simple since three components are involved.

Several factors determine the lifetime of electronic devices. In automotive electronics the most important damage mechanism is thermomechanical fatigue. Often the electronic devices are placed close to the engine of the car. The heat generated by the engine causes the materials to expand. Both are made from different materials, generally the circuit board expands more than the components. This causes a thermal mismatch. The situation is shown schematically in figure 4.3. The solder joints attach the component (e.g. a chip) to the board. If the board expands more than the chip, the joints have to deform to accommodate this mismatch. This causes stresses and plastic deformation inside the joints. Due to the high homologous temperatures the plastic deformation is mostly time-dependent creep.

Temperature variations occur frequently in cars. The temperature increases if the engine is running, the temperature decreases if the car is parked. The repeated creep deformation causes crack initiation, crack growth and ultimately results in the failure of the device because the electrical connection are interrupted. To guarantee a certain performance, car manufacturers demand that the electronic devices used in their vehicles can withstand a certain amount of temperature cycles of a given profile without loosing their functionality. The electronics industry performs thermal cycling tests



(a) Joint deformation at room temperature



(b) High temperature joint deformation

Figure 4.3.: *Solder joint deformation due to temperature variations*

to ensure that these requirements are met. However, since these test are time consuming and expensive, a reliable lifetime prediction methodology is desirable.

4.2. Standard methods for predicting the lifetime of solder joints

The lifetime prediction methodology currently employed in most industry applications can be divided into the following substeps:

- The material properties of the solder joint, the component and the board are determined experimentally. Suitable material laws are developed.
- A finite element model of the critical component including the joints and the board is generated. The temperature profile used in the accelerated tests is chosen as a boundary condition in the simulation and the stress-strain behavior of the joints during a thermal cycle is calculated. A quantity which is assumed to be related to the solder fatigue is extracted from the simulation.
- The lifetime of the solder joints undergoing accelerated thermal cycling is determined experimentally.
- The experimentally determined lifetime is correlated with the quantity extracted from the simulations using an empirical approach.

To perform the last step, experimental results involving the assembly in question are needed. The idea is of course to determine the correlation once and then to use it for different assemblies and temperature profiles, otherwise lifetime prediction would be pointless.
The materials forming the component and the circuit board are usually modeled using a linear elastic material description in connection with thermal expansion properties. The joints are assumed to be homogeneous materials with isotropic mechanical properties. However, due to the high homologous temperatures during thermal cycling creep deformation is known to occur

in the solder joints. Different models can be used to describe the creep behavior of the joints. One commonly used law to describe the (steady-state) creep is known as the double power law:

$$\dot{\epsilon}_{cr,s} = \frac{A_1}{T}\sigma^{n_1} \exp\left(\frac{-Q_1}{kT}\right) + \frac{A_2}{T}\sigma^{n_2} \exp\left(\frac{-Q_2}{kT}\right) \tag{4.1}$$

k is the Boltzmann constant, σ is the stress and T the temperature in Kelvin. The constants A_1, A_2, Q_1, Q_2, n_1 and n_2 have to be determined experimentally. The two exponential terms are often associated with different creep mechanisms. A version with only one exponential term is also often used.

Another possibility to describe the secondary creep is the hyperbolic sine law:

$$\dot{\epsilon}_{cr,s} = A_1(\sinh(A_2\sigma))^n \exp(-\frac{Q}{kT}) \tag{4.2}$$

Again A_1, A_2, n and Q have to be determined experimentally.

Different strategies can be found in the literature to determine the creep law. A very natural way to approach the task is to use SnAgCu bulk specimen and measure the deformation of the specimen if subjected to different temperatures and loading conditions. Once the creep laws shown above are fitted to the experimental data they can be generalized to the three-dimensional case.

Wiese et al. [44] used, next to other specimens, a dog-bone specimen to conduct creep tests. The dog-bone specimen is designed such that traction forces can easily be applied. The elongated shape of this specimen guarantees a homogeneous stress distribution, which is important because the data has to be fitted with respect to the stress state.

Other authors used specimens which were designed for applying shear forces [45]. One might argue that shear forces are the forces that actually appear in the field, and therefore the creep experiments should be based on them. On the other hand due to their shape these specimens are likely to introduce non-homogeneous stress states. Deplanque [46] realized this and designed a specimen which minimizes unwanted effects by optimizing its shape.

But the material properties of the solder joints in the electronic device may differ from those of the bulk specimen. The microstructure of SnAgCu is dependent on the cooling condition. The temperature gradient inside the

joints during solidification differs with the size of the sample and therefore the microstructure of the joints is different than the one found in bulk specimen. Knowing this, Wiese et al. [44] used actual flip chip solder joints to develop their creep law. Since the stress state is highly non-homogeneous, numerical simulation has to be used additionally in the process.
Several authors investigated the influence of primary creep [46,47] and found that this influences the outcome of the simulation. For a fixed temperature and stress state, primary creep is the initial creep behavior before a steady-state creep rate is reached. However, this will be neglected in this thesis.
To determine the reliability of solder joints experimentally, special test specimens are subjected to thermal cycling: The chips and circuit boards are designed such that the solder joints are connected in a series circuit. The resistance of the circuit is checked continuously during the experiments. A failure of one of the joints is recognized as an instantaneously increasing resistance. Schubert et al. [48] determined the lifetime of plastic ball grid array (PBGA) assemblies this way using temperature cycles from -40 to $+150$ degrees Celsius. Che and Pang [49] presented experimentally determined lifetimes of PBGA packages subjected to thermal cycles with a profile ranging from -40 to $+125$ degrees Celsius.
Another method to determine failure in solder joints can be applied to chip resistors. A shear force is applied to the resistor which is increased slowly. The force necessary to tear the resistor from the board is measured. If this force is significantly less than the force necessary to tear a freshly soldered resistor from the board a failure is assumed. Resistors which have been tested this way are of course not suitable for further thermal cycling. So the experiments have to be conducted with many resistors and a certain amount of them is removed at different times. Although this method is used frequently at Bosch, no discussion could be found in the literature.
The FEM is used to determine the structural behavior during thermal cycling. Data is gathered from a representative cycle. Due to the creep deformation of the joints the stress-strain curves show a hysteretic behavior. The stress-strain curves stabilize after several cycles, a cycle with a stable hysteresis loop is called a representative cycle.
Two different quantities are commonly used for correlation with experimental results. The accumulated creep strain and the creep strain energy

density.
The equivalent creep strain $\epsilon_{cr,eq}$ can be calculated from the creep strain tensor. If the absolute value of its rate is integrated over time, this gives the accumulated creep strain ϵ_{acc}. This is a measure for the cyclic creep deformation at a certain point in the solder volume.
The creep strain energy density W_{cr} is calculated as follows. The equivalent stresses and strains can be computed at every point of the solder volume. The creep strain energy density is the area enclosed by the hysteresis loop formed by the stress-strain curve of a representative cycle.
The quantity of interest is averaged over a solder joint which is assumed to be critical or over a volume V of a critical joint in which damage is assumed. Constants c_1 and c_2 are used to fit a lifetime law of one of the following forms:

$$N_f = c_1 \left(\frac{1}{|V|} \int_V \epsilon_{acc} \, dv \right)^{c_2} \tag{4.3}$$

$$N_f = c_1 \left(\frac{1}{|V|} \int_V W_{cr} \, dv \right)^{c_2} \tag{4.4}$$

N_f is the number of cycles to failure. Syed [50] used the accumulated creep strain and the creep strain energy density to determine fatigue laws for SnAgCu joints. Schubert et al. [51] also used both quantities to determine fatigue laws, which they used to predict the lifetime of SnAgCu solder joints in flip-chips.
Zhang et al. [45] presented a fatigue law for SnAgCu in which they used an energy-partitioning approach. In this approach not only quantities related to creep were used in the fatigue law, but also quantities related to time-independent plastic and elastic deformation. However, Syed [50] concluded that time-independent plasticity can be neglected in the model if thermal cycling is considered.

4.3. New methodology

To motivate a new methodology we will first discuss the shortcomings of the standard lifetime prediction as discussed in section 4.2. Afterwards the damage mechanisms are discussed from a physical point of view using the

literature currently available on that subject. From this discussion we may conclude, that in order to arrive at a better lifetime prediction the crack growth and the grain structure of the joint should both be represented in the simulation.

4.3.1. Motivation

The procedure outlined in section 4.2 has several disadvantages. The fatigue laws that are determined for a certain joint geometry and temperature profile may not be valid under different experimental conditions. Darveaux [52] noted that the lifetime prediction can differ by a factor of seven from experimental results if the fatigue law has not been determined under consistent experimental conditions. Therefore, if significant changes are made to the geometry or the temperature profile the lifetime prediction may give erroneous results.

But due to technological advances and changing customer demands the electronics industry constantly has to deal with changing components, different circuit boards and other joint geometries (e.g. ongoing miniaturization of electronics). Instead of continuously conducting new experiments the simulation methodology could be improved. If the damage mechanisms are included in the model, the simulation is a better description of the experimental reality. Therefore, one may hope that a simulation methodology based on such a model, is transferable to a larger class of experimental conditions once it is validated.

To understand the damage mechanism in solder joints we have to take a closer look at the microstructure. Many authors have investigated the microstructure of lead-free solder joints after the soldering process [53–60]. The structure consists of a dendritic β-Sn phase in which CuSn and AgSn particles are dispersed. The dendrites are relatively small compared to the joint size. The same is true for the CuSn and AgSn particles.

Tin has an anisotropic crystal structure. The mechanical behavior is therefore dependent on the crystal orientation. For pure tin the stiffness varies up to a factor of three depending on the orientation [61]. The crystal orientation of the dendrites is the same over large areas of the joint. These regions are mostly referred to as dendrite colonies or grains. The dendritic

structure of the tin matrix makes it hard to determine where different grain boundaries are located if optical microscopy is used. But using technologies like EBSD or polarized light microscopy, the solder joint can be subdivided in only a few different regions [56, 62] in which the crystal orientations are the same.

In the as-soldered state the joints only consist of a few grains [63]. Although recrystallization occurs in some areas [64–66] during thermal cycling, the recrystallized grains are still large compared to the joint measurements.

The crack spreads in a web-like fashion [64] along the grain boundaries [65, 67] during thermal cycling. Due to the anisotropy of the grains particularly unfavorable grain structures are found in about 10% of the joints which give rise to early failures [68].

Although the damage mechanism is not fully understood yet, these observations suggest that a crack growth simulation should be coupled with a model for the grain structure. Damage is generally related to the stresses and the plastic deformation inside a structure. The relative size of the grains and the highly anisotropic behavior of tin does not justify a calculation of these quantities based on a homogeneous joint description. Also, a homogeneous joint model could not predict web-like crack patterns. Furthermore, the early failures, which are a consequence of the non-homogeneity of the structure and its random nature, can only be reproduced if the highly anisotropic mechanical behavior of the grains is included in the simulation.

Several authors realized this and used the FEM to simulate joints formed by several grains. Gong et al. [69] simulated joints with two distinct grains as well as joints with a finer grain structure. The fine grain structure was modeled using cube-shaped finite elements of equal size, each element representing a different grain. Erinc et al. [70] simulated more complex grain shapes, but only two grains were used to model a joint. Damage was included by using a cohesive zone approach at the grain boundaries. Menk and Bordas [71] investigated the accumulated creep strain in the case of a joint formed by six distinct grains.

Other approaches to simulate cracks in solder joints without modeling microstructural features can be found in the literature too. Ghavifekr and Michel [72] calculated stresses in the vicinity of a crack tip which was artificially introduced to a homogeneous isotropic joint model with the FEM.

However, crack growth was not simulated. Towashiraporn et al. [73] simulated crack growth in a BGA joint using a homogeneous isotropic solder model and cohesive elements at the copper-solder interface. It was found that the simulated crack growth corresponds well with the experimentally determined crack growth. However, the solder alloy considered in their work was SnPb.

The microstructures in actual solder joints arise from a stochastic process. It is not possible to predict how the microstructure will form in a particular joint before soldering. To capture the variety of possible grain structures in a lifetime prediction it is proposed here to generate a set of random microstructures for the joint in question. Let us assume that the displacements in some area around the joint are known. In section 4.4.2 these displacements will be calculated from a global model of the device. The displacements can be used as boundary conditions for a crack growth calculation with a joint submodel. The lifetime of a particular joint, as determined by those crack growth calculations, depends on the microstructure. If many crack growth calculations for different microstructures but for the same joint are performed, it might be possible to make a statement about the failure probability. Such a statement would be much closer to the experimental reality than an estimated number of cycles to failure as determined by the standard methodology in section 4.2.

This idea raises several questions. One would have to find a way to generate realistic random microstructures for solder joints. Furthermore, constitutive laws describing the mechanical behavior of the microstructures are needed. A crack growth criterion based on these constitutive laws must be developed. Finally, an algorithm is needed which can perform these calculations for several thousand temperature cycles in a reasonable amount of time.

4.3.2. Grain structure generation

In section 4.4.2 joints in ball grid array (BGA) assemblies will be considered. Therefore, this section is concerned with the random generation of grain structures for the corresponding joint shapes. We are particularly interested in near eutectic SnAgCu BGA joints with a diameter of roughly

600μm, since detailed information about the crack growth is available [74]. The procedure may however be applied to other joint geometries as well.
As previously mentioned, it is known that near-eutectic solders form relatively large grains compared to their size [63,65,67] right after solidification. Sundelin et al. [65] found an average of 3 grains in joints with a diameter of 350μm, while Henderson et al. [67] found an average of 8 grains in joints with a diameter of 900μm.
This initial microstructure is not stable during thermal cycling, the upper part of the joint connected to the component recrystallizes [64–66] due to the high thermal strains in this region. No detailed discussion about the number of grains, their shape and the joint volume in which recrystallization takes place could be found in the literature. But evaluating the pictures published by Sundelin et al. [65] one notices that in their studies the recrystallized area is found in the upper fourth of the joint and contains roughly 30 grains in the cross-sectioned area.
Since the dendrites and the intermetallic dispersions inside the grains are small compared to the joint size [53–60], a homogeneous description of the grains themselves will be used in the simulation. To be able to construct joints with random grain structures we make the following assumptions for BGA joints with a diameter of 600μm:

1. Recrystallization occurs during the first few thermal cycles and happens instantaneously. The number of cycles until recrystallization occurs is negligible compared to the total lifetime of the joint.

2. The joint contains 6 grains on average after solidification.

3. The area that recrystallizes is the upper fourth of the joint. The recrystallized area contains 40 grains on average.

4. The initial as well as the recrystallized grain structure can be represented by a Voronoi tessellation applied to a random point distribution in the corresponding area.

5. The orientation of each grain is random and not related to the other grain orientations.

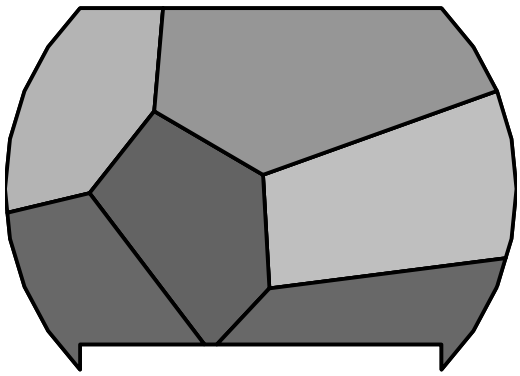
The first assumption is an idealization of the actual recrystallization process. The idea that recrystallization happens mostly during the first cycles

can be motivated by thermodynamical considerations, but its validity remains to be tested.
The second and the third assumption are a rough guess about the number of grains that was made based on the discussion at the beginning of this section. These assumptions are reasonable simply because similar observations have been made for joints of similar sizes. However, it should be mentioned that the number of grains may also depend on other factors like the position of the joint in the assembly.
The fourth assumption can be motivated for the intial microstructure. The Voronoi tessellation and related concepts are often used in materials science to generate random grain structures [75–77]. It starts with a random source point distribution. A polygon is generated for each source point containing exactly those points in $\mathbb{R}^2$ or $\mathbb{R}^3$ which are closer to this particular source point than to all the others. The tessellation can therefore be used to model the grain structures of solidified metal alloys. Solidification also starts from randomly distributed solidification sources in the liquid phase. Each solidification source grows and consumes the liquid phase until it reaches another source. However, recrystallization is based on other mechanisms and the grain shapes may differ from the ones generated by Voronoi tessellations.
The fifth assumption is made because no detailed studies about the grain orientations were found in the literature which allow further conclusions. Twinning may sometimes occur [63], in these situations the orientation in different grains would be related to each other.
Based on these assumptions the following procedure is proposed to generate random grain structures:

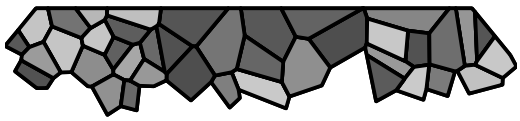
1. Generate an as-solidified grain structure based on a Voronoi tessellation using 6 randomly distributed source points in the solder volume. Assign a random orientation to each of these grains.

2. Generate a recrystallized grain structure based on a Voronoi tessellation using 40 randomly distributed source points in the upper fourth of the solder joint. Assign a random orientation to each of these grains.

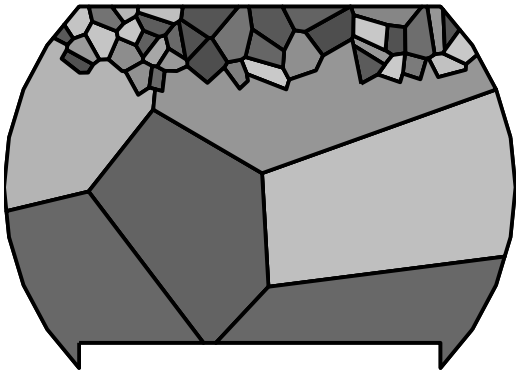
3. Superimpose the recrystallized structure onto the as-solidified structure.



(a) Initial grain structure



(b) Recrystallized area in the upper joint region



(c) Final model of the joint

Figure 4.4.: *Generation of random recrystallized grain structures for a solder joint*

As an example one may consider the structure shown in figure 4.4. The boundary of the initial structure in 4.4(a) describes a two-dimensional model of a solder joint which forms if BGA devices are soldered onto a circuit board. The flat part on the top is attached to a copper pad. The cavity on the lower part of the joint encloses another copper pad.
A recrystallized structure forming in the upper part of the joint is shown in figure 4.4(b). The final joint model in 4.4(c) is obtained by superimposing the recrystallized area onto the initial grain structure.

4.3.3. Crack growth criterion

The equivalent creep strain $\epsilon_{cr,eq}$ can be calculated from the creep strain tensor. If the absolute value of its rate is integrated over time, this gives the accumulated creep strain:

$$\epsilon_{acc} := \int_0^T |\frac{\partial \epsilon_{cr,eq}}{\partial t}| dt \tag{4.5}$$

This is a measure for the creep that has taken place at a certain point in the solder volume. The quantity is closely related to the solder fatigue process and is often used for lifetime prediction as already discussed in section 4.2. We will use it in a different way.
During thermomechanical fatigue the crack propagates along the grain boundaries [65,67]. Classical creep crack growth along grain boundaries starts with small cavities which grow and coalesce to form a macroscopic crack [78]. The cavity formation and growth is an accumulation of voids over time. Creep generally involves void movement through the crystal or the grain. Thus, the creep deformation in the vicinity of the boundary is related to the failure of the boundary.
A cracking criterion for a particular grain boundary can therefore be proposed as follows. An area A_B surrounding the boundary B is determined which is formed by all points with a distance smaller than d_B to the boundary B. The situation is shown in figure 4.5. The boundary cracks if:

$$\epsilon_{avg,B} := \frac{1}{|A_B|} \int_{A_B} \epsilon_{acc}\ da > c_{cr} \tag{4.6}$$

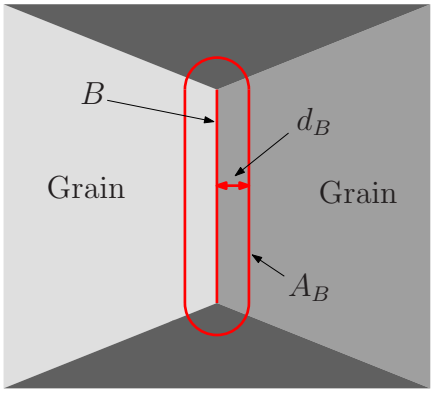


Figure 4.5.: *Area A_B around a boundary B used in the cracking criterion*

$|A_B|$ is the size of the area and the constant c_{cr} must be chosen appropriately. Thus, in the implementation boundaries will always crack as a whole, which is an approximation to the experimental situation.
The choice of d_B is not easy to justify. The idea is to determine an area A_B in which void movement inside the crystal significantly contributes to cavity formation along the boundaries. In section 4.4.2 a value d_B will be chosen such that it is smaller than most of the grains, but such that A_B still contains a reasonable area around each boundary.

4.3.4. Crack growth algorithm

The random generation of grain structures together with the crack growth criterion and the constitutive laws for the grains can be used to predict the crack growth in a particular joint. But performing this simulation for all temperature cycles would be very time-consuming. However, one can assume that if the geometry of the joint does not change, the creep strain that accumulates during each cycle is the same if the system is in a steady state (i.e. if the deformation of the joint in two subsequent cycles is roughly the same).
Due to the nonlinearity of the creep law the stress-strain curves evaluated for points inside the solder form hysteresis loops. For a fixed geometry it takes several cycles until these loops stabilize. The first cycle in which a steady-state behavior can be assumed is called a representative cycle. The accumulated creep strain ϵ^1_{acc} is calculated at every Gauss point by evaluating (4.5) for such a cycle.

If we are at the beginning of a crack growth calculation the crack criterion (4.6) can be replaced by:

$$\frac{N}{|A_B|}\int_{A_B}\epsilon^1_{acc}\,da > c_{cr} \tag{4.7}$$

Equating both sides, we can find the number of cycles N after which the boundary cracks. This criterion is evaluated for every boundary. The number of cycles N^1_{min} after which the first boundary cracks is the number of cycles during which ϵ^1_{acc} may be used to calculate the creep behavior. If the first boundary is cracked the geometry has to be updated which changes the creep behavior. The new criterion for a cracking of a boundary B after determining ϵ^2_{acc} from another representative cycle becomes:

$$\begin{aligned} & N^1_{min}\tfrac{1}{|A_B|}\textstyle\int_{A_B}\epsilon^1_{acc}\,da\ \cdots \\ + \quad & N\tfrac{1}{|A_B|}\textstyle\int_{A_B}\epsilon^2_{acc}\,da > c_{cr} \end{aligned} \tag{4.8}$$

Evaluating N for all boundaries gives another number of cycles N^2_{min} after which the second boundary fails. More generally in the i-th step for every boundary B one has to evaluate:

$$\begin{aligned} & \textstyle\sum_{j=1}^{i-1} \quad N^j_{min}\tfrac{1}{|A_B|}\textstyle\int_{A_B}\epsilon^j_{acc}\,da\ \cdots \\ + \quad & N\tfrac{1}{|A_B|}\textstyle\int_{A_B}\epsilon^i_{acc}\,da > c_{cr} \end{aligned} \tag{4.9}$$

The minimum over all values of N gives the number of cycles N^i_{min} after which the next boundary fails. Updating of the geometry is done until the joint fails, i.e. until the crack has propagated through the whole joint. For reasons of computational efficiency, cracks are introduced at all boundaries that fulfill the criterion (4.9) with a value of $0.95c_{cr}$ on the right hand side and $N := N^i_{min}$. This ensures a reasonable progress for fine grain structures. Furthermore, we prevent the grain boundaries at the copper-solder interface from cracking. The copper dissolves into the solder [79] which results in an interface that is tougher than the interface between different SnAgCu grains. The algorithmic procedure for calculating crack growth in a solder joint with a given grain structure is visualized in figure 4.6.

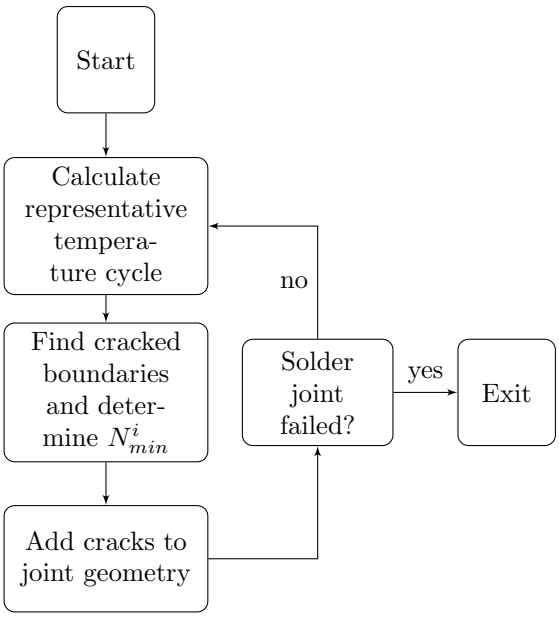


Figure 4.6.: *Algorithmic procedure to determine crack growth in solder joints*

4.4. Numerical examples

In this section numerical examples are presented to demonstrate the application of the concepts discussed so far to solder joints. To apply the crack growth algorithm, constitutive laws for the grain structures are needed. Those will be determined in section 4.4.1 in an inverse procedure. Using these material parameters crack growth calculations for solder joints are performed in section 4.4.2.

All of the strategies discussed so far have been used to solve the structural problems in a fully automated procedure.

The numerically determined enrichment functions have only been applied to crack tips and to the re-entrant corners which the joints in section 4.4.2 form with the substrate and the board.

In order to apply the preconditioning technique the structural domains were decomposed into three (structures in section 4.4.1) and six (structures in section 4.4.2) strips of equal width.

The thermal expansion in the z-direction was set to zero. Otherwise, due to the plane strain assumption, an increased temperature would generate stresses even if no forces or displacement boundary conditions are imposed onto the structure. The reference temperature for all materials was set to 295 K.

Meshes were generated using the procedure described in section 3.4. n_{gr} was set to 4 and n_b was set to 10. The boundaries or interfaces connected to the re-entrant corners at the top and at the bottom of the joints in section 4.4.2 have been refined with a value of $n_b = 20$. The initial mesh for the rectangular structures in section 4.4.1 was the same as the mesh in figure 3.40(a). For the examples in section 4.4.2 the initial mesh consisted of equally sized triangles similar to those in figure 3.40(a) with an initial edge length of about 20 μm.

To determine the displacement rates in section 4.4.1 a time step of $\Delta t = 1$ s was chosen. For the crack growth simulations in section 4.4.2 time steps of $\Delta t = 10$ s were found to be a better balance between computation time and accuracy.

For all grain boundaries the value $d_B = 0.0025$ mm was chosen to obtain an area A_B for the determination of the averaged accumulated creep strain

A [s^{-1}]	n	Q [kJ/mol]	σ_n [MPa]	R [kJ/mol]
1e-11	12	74.8	1	0.008314

Table 4.1.: *Creep constants as published by Wiese et al. [44]*

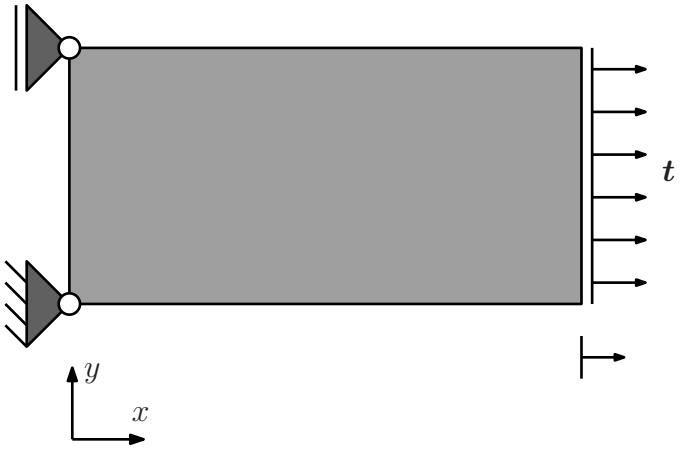
as in figure 4.5.
Due to the complicated crack patterns which may form during crack propagation, it is possible that some grains are completely disconnected from the rest of the structure. Those grains have to be detected and removed.

4.4.1. Determination of material parameters

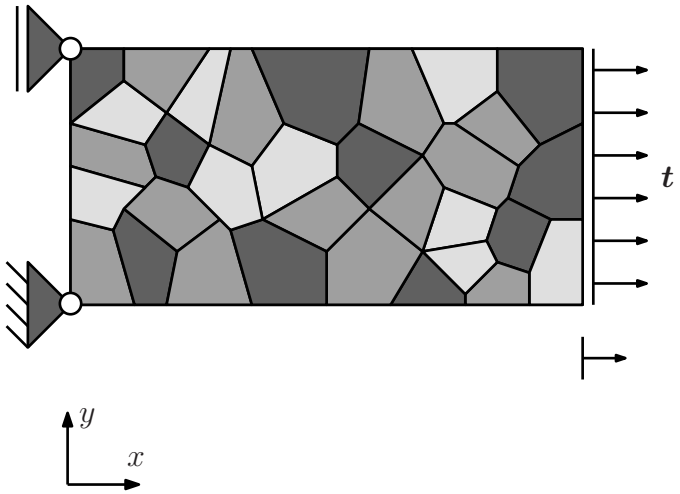
We are interested in modeling the mechanical behavior of SnAgCu grains with an alloy composition of 95.5 wt% Sn, 4 wt% Ag and 0.5 wt% Cu (SAC405). If a solder specimen contains a large number of grains one may assume that the structure is homogeneous and that it can be modelled using an isotropic material description. In that case the Young's modulus of SAC405 may be set to 41 GPa and the Poisson's ratio to 0.3 [44]. A reasonable coefficent of thermal expansion for such a model would be 20 ppm/K [48]. An appropriate creep law of the following form was determined by Wiese et al. [44]:

$$\dot{\epsilon}_{cr} = A \left(\frac{\sigma_{vm}}{\sigma_n} \right)^n \exp\left(-\frac{Q}{RT[\mathrm{K}]} \right) \tag{4.10}$$

$T[\mathrm{K}]$ is the temperature measured in Kelvin. The other parameters are shown in table 4.1. The key idea in this section is that the material models describing the mechanical behavior of the grains should represent the mechanical behavior of the homogeneous model on a large scale.
The situation is visualized in figure 4.7. If traction forces are applied to the homogeneous solder model in figure 4.7(a), SnAgCu grain structures, such as the one shown in figure 4.7(b), should on average react in the same way to these tractions, that is, the displacement on the right-hand side should be the same. This should also be true for the creep rate at different temperatures.



(a) Homogeneous isotropic solder model



(b) Solder model formed by grains

Figure 4.7.: *Deformation of a homogeneous isotropic solder model compared to the deformation of a solder model formed by a number of SnAgCu grains*

Assume that the homogeneous structure in figure 4.7(a) has a length of 2 and a height of 1, the units themselves are not relevant. For different temperature increases ΔT (with respect to the reference temperature) and tractions, the displacements and displacement rates at the right-hand side have been determined. The results are shown in table 4.2. The tractions

Elastic displacement 0.5 GPa	0.022195
Thermal expansion $\Delta T = 100$ K	0.0052
Displacement rate, $\Delta T = 0$ K 20 MPa	4.0936e-10 s^{-1}
Displacement rate, $\Delta T = 0$ K 40 MPa	1.6767e-6 s^{-1}
Displacement rate, $\Delta T = 40$ K 20 MPa	0.0002083 s^{-1}

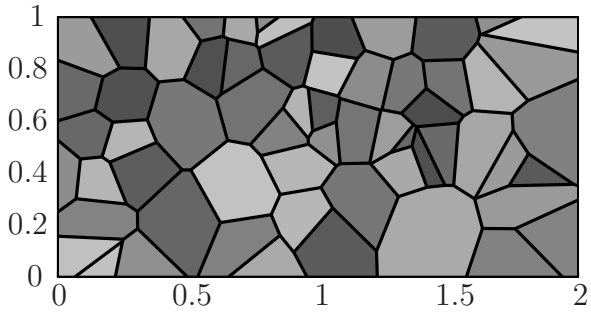
Table 4.2.: *Deformation of the homogeneous solder model for different loading conditions*

used to determine the displacement rates are similar to the experimental conditions for which the creep law was determined. The displacement rates have been determined by calculating the displacements at $t = 0$ s and $t = 1$ s and dividing the difference by 1 s.

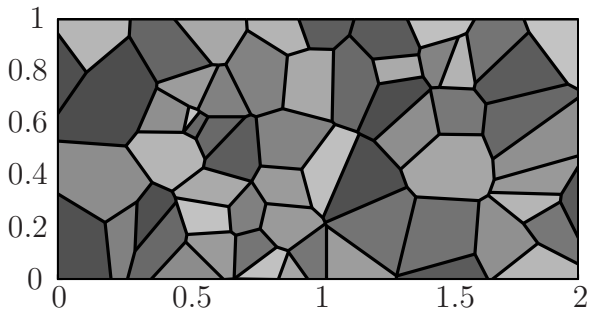
The material laws for the grains will be determined such that they match the mechanical behavior of the homogeneous model for these loading conditions. Since the creep deformation of any structure is dependent on the stresses and the temperature, the elastic and thermal properties of the grains have to be determined before the transient behavior is considered.

A series of grain structures was generated by Voronoi tessellations which are shown in figure 4.8. The structures have the same proportions as the homogeneous joint model used to determine the values in table 4.2. A random orientation is assigned to each of the grains.

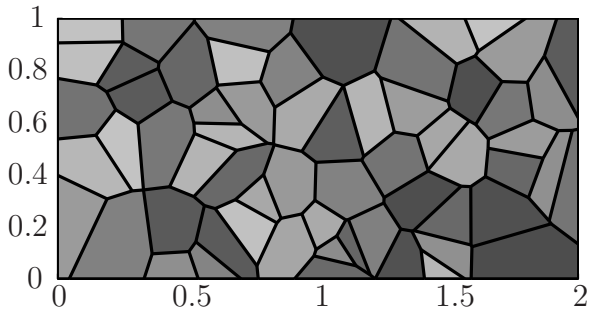
The elastic constants for tin grains were determined by Mason and Bommel [80] and are shown in table 4.3. The elasticity tensor $\boldsymbol{D}_{Sn}$ for arbitrary grain orientations can be determined by applying the correct transforma-



(a) Structure 1



(b) Structure 2



(c) Structure 3

Figure 4.8.: *Different structures formed by grains with a random orientation used to determine constitutive laws for SnAgCu grains*

D_{11}	D_{12}	D_{13}	D_{33}	D_{44}	D_{55}
73.5	44.2	28	87	22	22.65

Table 4.3.: *Tin elastic constants ([GPa]) by Mason and Bommel [80] in Voigt notation*

[100]	[010]	[001]
15.4	15.4	30.5

Table 4.4.: *Tin thermal expansion ([ppm/K]) measured for different crystal directions by Yang and Li [81]*

tion. The same is true for the coefficients of thermal expansion which were determined by Yang and Li [81] as shown in table 4.4. It is assumed that the elastic behavior of the SnAgCu grains is similar to that of the tin grains and that the elasticity tensor of a SnAgCu-grain can be approximated by:

$$\boldsymbol{D}_{SnAgCu} \approx c_{el}\boldsymbol{D}_{Sn} \tag{4.11}$$

This assumption is reasonable since the SnAgCu-alloy consists mostly of tin.

Setting $c_{el} = 1$, the displacements shown in table 4.5 are obtained for the structures in figure 4.8. Comparing this with the displacement of the homogeneous solder model, we obtain an optimal value for c_{el} as:

Structure 1	0.01563
Structure 2	0.01565
Structure 3	0.01574

Table 4.5.: *Elastic deformation of the polycrystalline structures in figure 4.8 using a force of 0.5 GPa and* $c_{el} = 1$

$$c_{el} = 0.7029 \tag{4.12}$$

The right hand side displacement due to thermal expansion of the grain structures is given in table 4.6. Comparing this with the thermal expansion of the homogeneous model, we may conclude that the coefficients of thermal expansion for the tin grains provide useful values to describe the thermal expansion of the SnAgCu grains if the behavior on the large scale is considered.

For each of the three load cases in table 4.2 in which displacement rates are considered, a displacement rate for the structures in figure 4.8 can be calculated if creep constants are known. Due to the non-homogeneity of the

Structure 1	0.0051618
Structure 2	0.0052575
Structure 3	0.0051056

Table 4.6.: *Displacement of the polycrystalline structures in figure 4.8 due to thermal expansion using a temperature increase of* $\Delta T = 100\ K$ *and the thermal expansion properties of tin grains*

polycrystalline structures, the displacement rates should be averaged over a time interval to obtain a representative value. A time interval of 10 s was chosen here. For the same reason the displacement rates were also averaged over all nodes along the free vertical edge of the structure.
Depending on the creep constants, the displacement rates calculated for the structures in figure 4.8 will differ from those calculated for the isotropic structure. This difference is a function of the constants determining the creep law.
Summing up the absolute value of these differences for all structures and load cases, we can define an error for given creep constants. A Newton-type method was used to find creep constants which minimize this error. The final constants are summarized in table 4.7. σ_n and R were held constant

A [s^{-1}]	n	Q [kJ/mol]	σ_n [MPa]	R [kJ/mol]
1.08e-11	11.04	71.06	1	0.008314

Table 4.7.: *Adjusted creep constants for SnAgCu grains*

during the minimization. The corresponding displacement rates are summarized in table 4.8. Deviations can be seen if the values are compared to the creep rates of the isotropic structure in table 4.2, but taking into account that the displacement rates range over several orders of magnitude one may conclude that the creep law determined here is useful for our purposes.

4.4.2. Crack growth calculations

To demonstrate the functionality of the methodology outlined in this thesis, crack growth calculations for a BGA assembly with a realistic geometry are

	ΔT=0 K 20 MPa	ΔT=0 K 40 MPa	ΔT=40 K 20 MPa
Structure 1	4.08e-10	8.6e-7	0.000208
Structure 2	1.9e-10	4.57e-7	0.00021
Structure 3	1.53e-10	3.2e-7	0.000192

Table 4.8.: *Displacement rates of the polycrystalline structures in figure 4.8 using the creep law in table 4.7 ($[s^{-1}]$)*

performed. A BGA is a flat quadratic package containing a chip. When attached to the circuit board by soldering, the solder joints form little balls below the package. The position of the balls below the chip in a PBGA 676 package assembly is shown in figure 4.9. To obtain a two-dimensional

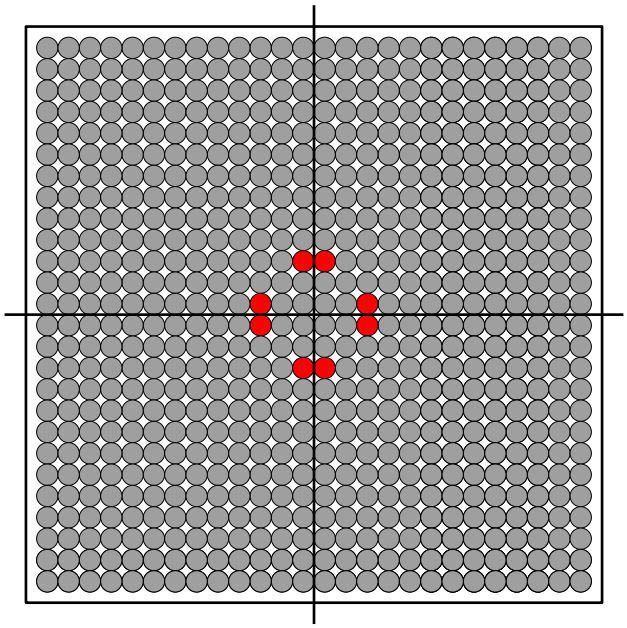
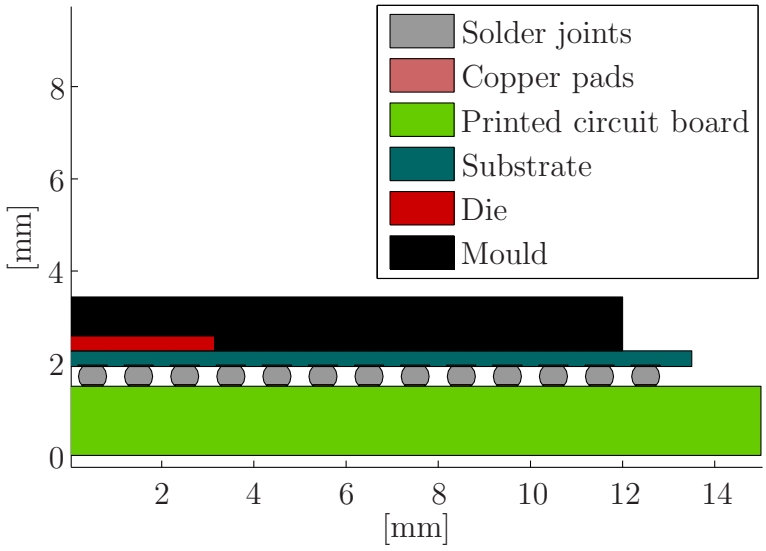
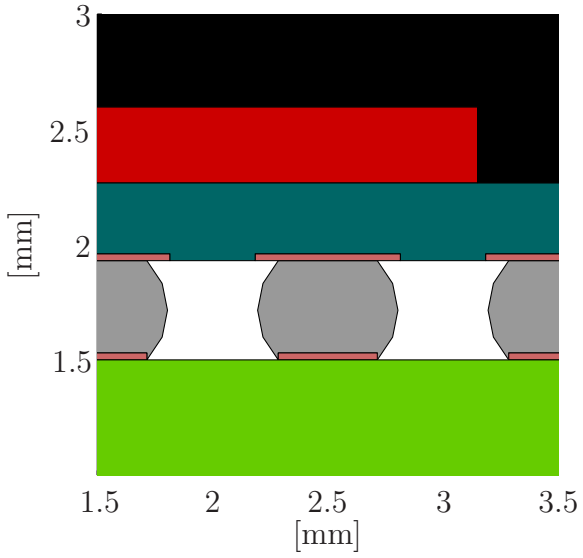


Figure 4.9.: *Position of the solder balls below the chip in a PBGA 676 package and lines of symmetry (red ball positions are those for which crack growth data was gathered by Tunga [74])*

model of the assembly, we think of an either vertical or horizontal cut along a line of balls close to the center (e.g. the line of balls above the horizontal symmetry line). Exploiting the symmetry, the device can be modeled as shown in figure 4.10(a). The third joint position from the left is magnified in figure 4.10(b). Due to the symmetry of the device, this corresponds to



(a) Global Model



(b) Magnification of the third joint from the left which is used in a submodel to calculate crack growth

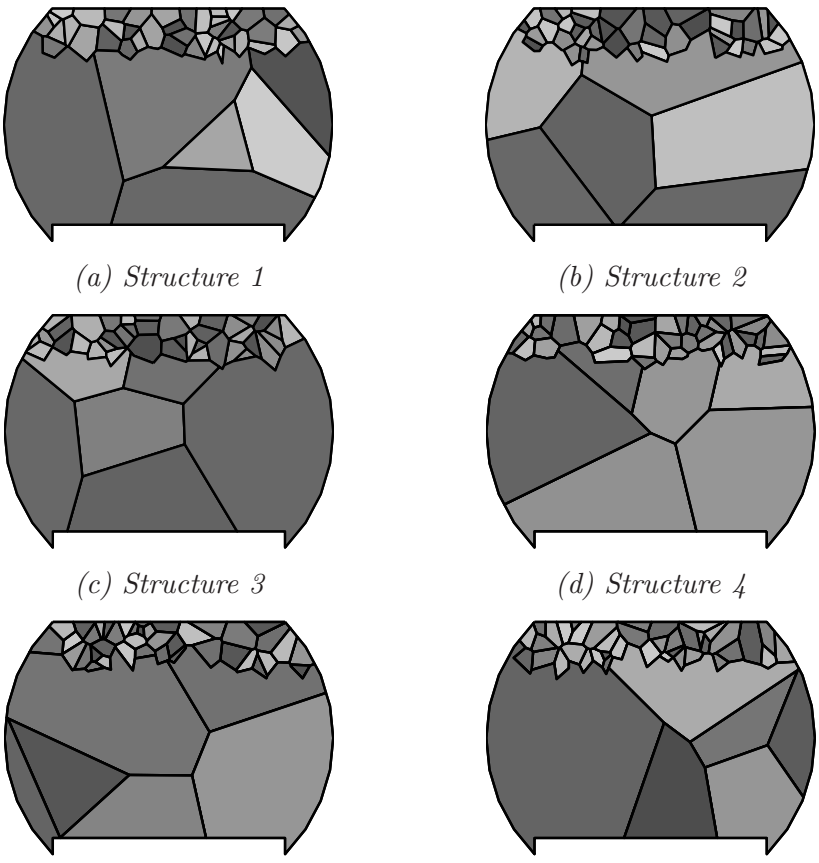
Figure 4.10.: *Model of the PBGA 676 package soldered onto a circuit board*

any of the ball positions in figure 4.9 marked in red.
Tunga [74] measured the crack growth in such an assembly for two solder ball positions. The SAC405 solder alloy was used for soldering. The ball magnified in figure 4.10(b) is one of the ball positions for which sufficient experimental data was gathered by him (although the actual lifetime was not determined for this ball position, we anticipate that if the simulation is able to reproduce the crack growth appropriately, the same is true for the lifetime).
The geometry in figure 4.10(a) will be used as a global model in the simulation. The corresponding material parameters are shown in table 4.9. Calculations were done with the global model using homogeneous joint mod-

	CTE [ppm/K]	E [GPa]	ν
PCB	13	22.4	0.15
Substrate	15	28.5	0.3
Die	2.8	131	0.3
Mould	9	25	0.3
Copper pads	17.3	121	0.3

Table 4.9.: *Material properties of the PBGA 676 package assembly*

els described by the corresponding material laws as already used in section 4.4.1. Zero displacements in the horizontal direction were prescribed at $x = 0$ and in horizontal and vertical direction at $(0, 0)$. The displacements along the upper boundary of the upper copper pad and along a line 30 μm below the lower copper pad of the joint in figure 4.10(b) were used in a submodel. Randomly generated grain structures were used to model the solder joint in this submodel. They are shown in figure 4.11.
We want to address the effect of the enrichments without creep. Consider the grain structure in figure 4.11(a). The stress state was evaluated in different areas of the structure for an instantaneously increased temperature from 295K to 395K. Creep does not influence the deformation because the stresses are evaluated at $t = 0$. The areas which were examined more closely are shown in figure 4.12. A crack was artificially introduced in the upper right part of the structure.
The mesh generated for this structure contained 6803 finite elements. Parts



(a) Structure 1 *(b) Structure 2*

(c) Structure 3 *(d) Structure 4*

(e) Structure 5 *(f) Structure 6*

Figure 4.11.: *Grain structures used to model the BGA ball in the submodel*

of the mesh are shown in figure 4.13(b), figure 4.13(d) and figure 4.13(f). The von Mises stresses in the area around the re-entrant corner in the upper left of the joint are shown in figure 4.13(a). The large stress gradients

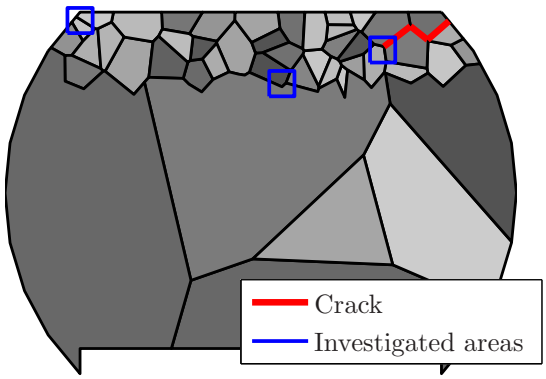
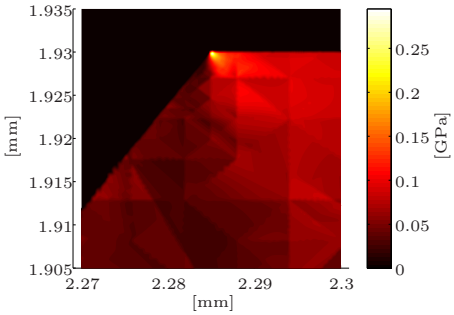


Figure 4.12.: *Areas investigated to address the effect of the enrichment functions*

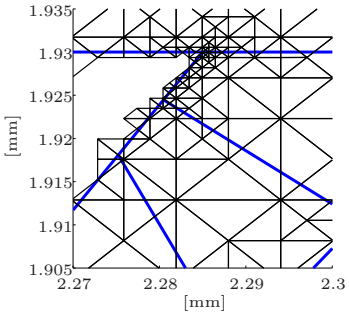
expected in the vicinity of the re-entrant corner are well resolved by the enrichments. Please note that the mesh around the re-entrant corner connects the outer part of the copper pad and the outer part of the solder ball. Thus, without the numerically determined enrichment functions, the deformation of both boundaries would be coupled by the finite element mesh outside the structural domain. The von Mises stresses around the second area closer to the center of the joint are shown in figure 4.13(c). The discontinuous stresses are well resolved although the grain boundaries do not align with the element edges. The von Mises stresses around the crack tip are shown in figure 4.13(e). The singular stresses around the crack tip are resolved very well by the enrichments. The structural part on the right shows only very low von Mises stresses. This indicates that the crack is accurately represented by the enrichments, since that part of the structure is relaxed by the presence of the crack.

According to equation (4.7) there is a linear relation between c_{cr} and the number of cycles after which a certain boundary fails. Tunga [74] found that crack growth for the joint in figure 4.10(b) generally starts after 810 thermal cycles. c_{cr} will be chosen such that this experimental result is reproduced on average.

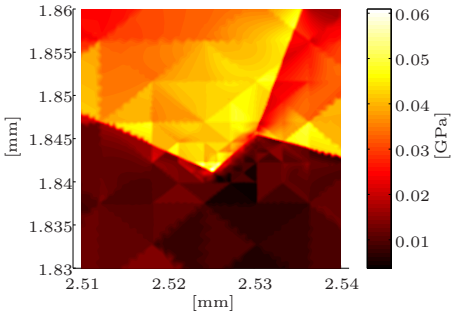
Therefore, the deformation of each structure needs to be calculated during a representative temperature cycle. The temperature profile used in the sim-



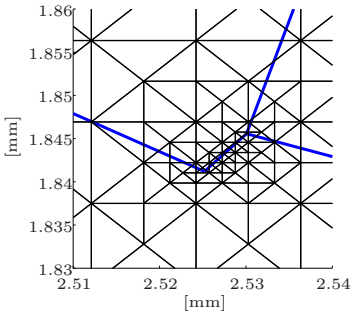
(a) von Mises stresses in the upper left corner of the joint



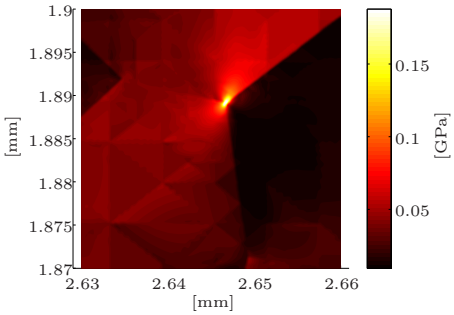
(b) Mesh in the upper left corner of the joint



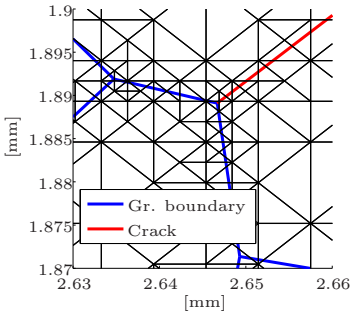
(c) von Mises stresses around a boundary close to the substrate



(d) Mesh around a boundary close to the substrate



(e) von Mises stresses around the crack tip



(f) Mesh around the crack tip

Figure 4.13.: *Meshing and von Mises stress distribution in different areas of the polycrystalline solder joint model due to an instantaneous temperature increase*

ulations is a reproduction of the temperature cycle used in the experiments performed by Tunga [74] and is shown in table 4.10. We will assume that

Time [s]	Temp. [K]
0	295
100	395
700	395
1100	215
1700	215
1800	295

Table 4.10.: *Temperature profile used in the simulation*

the second temperature cycle is a representative cycle. Taking the maximum of the averaged accumulated creep strain $\frac{1}{|A_B|}\int_{A_B}\epsilon^1_{acc}\,dA_B$ over all grain boundaries of the structures in figure 4.11 we obtain the values shown in table 4.11. A crack growth constant which ensures crack initiation after

Structure 1	0.006
Structure 2	0.003
Structure 3	0.0047
Structure 4	0.0037
Structure 5	0.0088
Structure 6	0.0046

Table 4.11.: *Maximum value of the averaged accumulated creep strain in the undamaged joint taken over all grain boundaries of the polycrystalline joint models*

810 cycles on average for the joint models considered here is therefore:

$$c_{cr} = 4.158 \tag{4.13}$$

The creep strain that accumulates during a representative cycle in structure 1 before crack initiation is shown in figure 4.14. The region in the upper left corner of the joint is pictured. Clearly, the accumulated creep strain distribution takes its highest values in the vicinity of the re-entrant corner.

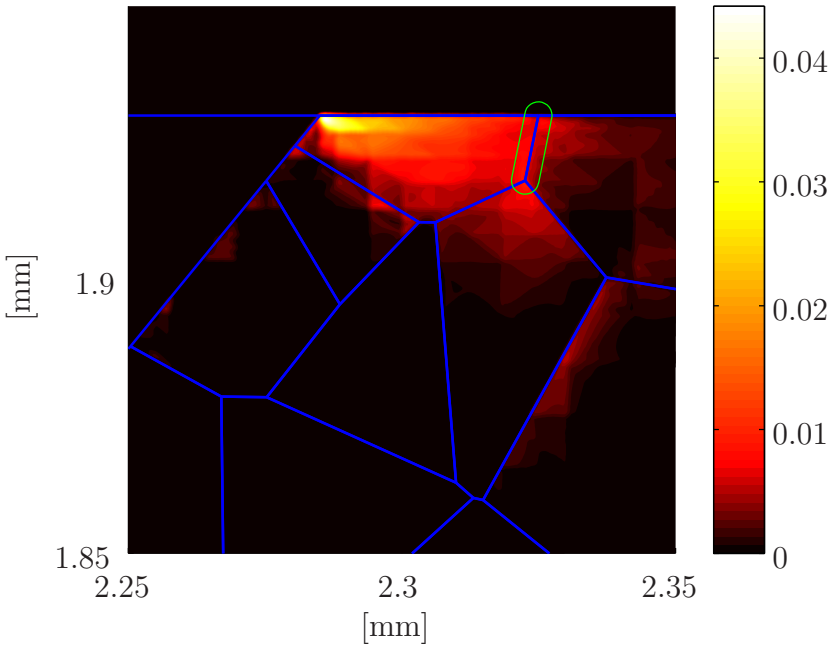


Figure 4.14.: *Accumulated creep strain distribution for a representative temperature cycle of structure 1 in figure 4.11 before crack initiation*

This is a consequence of the stress singularity that forms there. Another apparent feature that can be observed is that the distribution is discontinuous along some of the grain boundaries. Displaying this discontinuity numerically is only possible because the enrichment functions enable us to model discontinuous strains and stresses along the grain boundaries.
Since we assume that the interface between the copper pad and the solder cannot break, a boundary in the solder bulk fails first. The boundary is enclosed by a green line in figure 4.14. The line marks the area A_B used to evaluate $\epsilon_{avg,B}$ in equation (4.6). The crack growth patterns in structure 1 at different stages of crack growth are shown in figure 4.15. The patterns form a web-like structure like those seen in actual solder joints after thermal cycling [64]. The main crack initiates in the upper left part of the joint and propagates along the upper solder-copper interface.
We want to compare the crack growth in all structures with experimentally determined crack growth published by Tunga [74]. The comparison is shown in figure 4.16. For each cycle number in figure 4.16 at which the crack length has been determined experimentally, twelve different joints were in-

(a) Crack pattern after 678 Cycles

(b) Crack pattern after 1183 Cycles

(c) Crack pattern after 1892 Cycles

(d) Crack pattern after 2698 Cycles

Figure 4.15.: *Crack development in structure 1 from figure 4.11 (cracks are marked by red lines)*

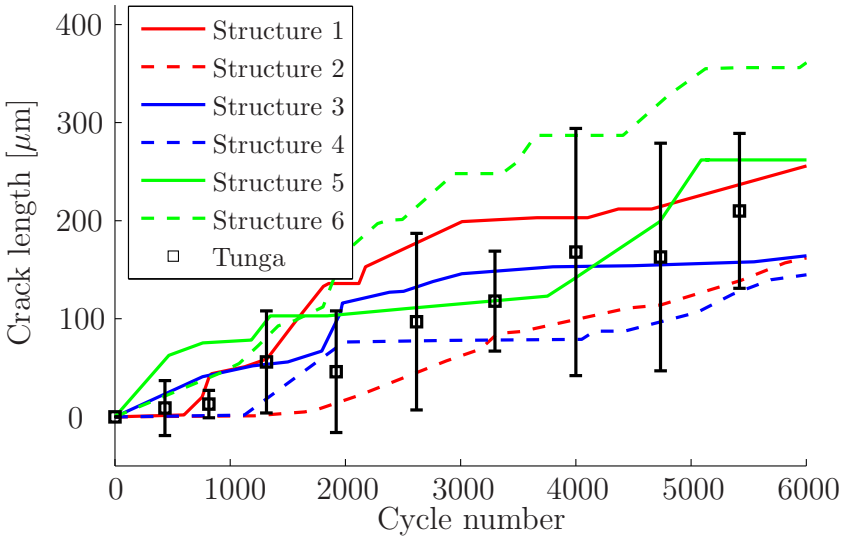


Figure 4.16.: *Crack lengths calculated for the polycrystalline solder joint models compared with data from the literature*

Cycle number	Avg. crack length by experiment [μm]	Avg. crack length by simulation [μm]	Relative error in crack length	Standard deviation by experiment [μm]	Standard deviation by simulation [μm]	Relative error in standard deviation
434	9	17	0.89	28	20	0.29
816	13	33	1.54	14	26	0.86
1315	56	51	0.09	52	33	0.37
1921	46	94	1.04	62	43	0.31
2618	97	126	0.3	90	57	0.37
3298	118	146	0.24	51	61	0.2
4000	168	160	0.05	126	69	0.45
4736	163	183	0.12	116	74	0.36
5421	210	213	0.01	79	80	0.01

Table 4.12.: *Comparison of mean crack length and standard deviation in experiment and simulation*

vestigated to obtain the standard deviation and the mean value [74]. Due to the different grain shapes and orientations used in the simulations for each joint model, different crack lengths are obtained. The standard deviation in crack length obtained from the experiments is visualized by the black bars in figure 4.16. It should be noted that the bars reach below the horizontal axis of the graph and that their length is not monotonically increasing with the number of cycles. Furthermore, the mean value of the crack length is also not monotonically increasing with the number of cycles. A possible reason could be the number of measurements used to determine the crack lengths in the experiments. We may conclude that additional measurements are necessary to obtain more representative values for the mean crack length. The same is true for the standard deviation.

The experimentally determined mean crack length and standard deviation is listed in table 4.12 for different cycle numbers as published by Tunga [74]. The mean crack length and the standard deviation was also calculated for the joint models used in the simulation. The mean crack length for the

first 2000 cycles differs significantly in experiment and simulation. But for increasing cycle numbers the mean crack length obtained by the simulation is a good approximation of the experimentally determined value.
The standard deviation obtained by the simulation is roughly 30% lower than the standard deviation obtained from the experiments for large cycle numbers. Increasing the number of measurements could help to obtain more representative values which may be closer to the ones determined by the simulation.
One might also argue that the number of grain structures used in the simulation is not sufficient to obtain representative values. But note that from the simulation we have obtained a mean crack length and a standard deviation which are both monotonically increasing with the cycle number.
Another reason for the differences in experiment and simulation could be the simplified description of the experimental reality in the model. Including the development of the grain structure over time could improve the solder model. Also the constitutive laws obtained by the inverse simulation are acceptable from a global point of view, but we cannot guarantee that they are a good description of the deformation process at the grain level.
However, the mean crack length and the standard deviation obtained by the simulation are still a good guess for the experimental values. Thus, we may conclude that the model proposed here is a good candidate to represent the crack growth in real solder joints by simulation.

5. Conclusions

The X-FEM is a powerful tool to solve problems in structural mechanics. Especially if the exact solution contains weak singularities, the approximation properties can significantly be enhanced by the use of enrichment functions. The shape of these singularities is dependent on the surrounding materials. In this thesis the idea of enriching the function space with weakly singular functions has been generalized to a wider class of problems by using numerically determined enrichment functions.

The numerical experiments that have been performed indicate that optimal convergence rates can be recovered for plane strain problems in linear elasticity by using numerically determined enrichment functions to represent weak singularities. For transient problems involving creep, however, a correct representation of the singularities cannot be expected. Further research should be done to develop an enrichment procedure which results in an accurate representation of weak singularities in creeping structures.

The numerically determined enrichment functions were used in combination with other enrichment functions to form a framework in which general polycrystalline structures containing cracks can be simulated with elements whose shape is independent of geometrical features. Due to the large number of enrichment functions that must be used, well-conditioned equation systems cannot be guaranteed any more. It was shown in this thesis, that in certain situations arbitrarily ill-conditioned equation systems can be the result of an enrichment procedure.

To resolve this problem a preconditioning technique was developed and tested. The technique employs a domain decomposition. Numerical experiments were conducted to test the performance. They indicate that even for extremely critical cases the preconditioner restores a condition number

close to the condition number of the FEM-stiffness matrices without any enrichment. As a result the equation systems for one of the example structures could be solved twice as fast.
It was furthermore discussed why the domain decomposition is a useful tool to minimize the computational effort for the calculation of the preconditioner.
The domain decomposition has another advantage. In a naive approach to improve the condition number one might attempt to solve the problem by applying a Cholesky-decomposition to the whole submatrix of enriched degrees of freedom. However, if the number of enriched degrees of freedom is large, the Cholesky-decomposition might break down due to roundoff errors. Because of the domain decomposition the Cholesky-decomposition has to be applied only to smaller submatrices. The Cholesky-decomposition is less likely to break down if the matrices are small because less floating point operations have to be performed.
In the future it would be nice to have a mathematical theorem which guarantees for arbitrary enrichments that the preconditioner is able to restore well-conditioned matrices. Such a theorem could exploit the fact that the enrichment functions (as opposed to the FEM shape functions) are zero at all nodes and that they are smooth inside the structural domain.
The enrichment procedure and the solution procedure by preconditioning have been combined with an efficient meshing scheme for polycrystalline structures. The advantage of this meshing procedure is that it can be automated completely. This is possible because the elements do not have to conform to geometrical features of the structure. Meshing is often seen as the bottleneck in the automation of simulation. However, the meshing procedure introduced here allows for a completely automated simulation if combined with the X-FEM. An element cluster was used to make the mesh refinement more efficient.
A penalty method was used to prevent crack faces from overlapping. The Woodbury matrix identity was used to combine the penalty method with the preconditioner.
This methodology has been used to calculate crack growth in solder joints based on microstructural phenomena. A special treatment of elements in which high creep rates appeared was necessary.

The grain structures for the solder joints have been randomly generated using Voronoi-tessellations. This random procedure has been adjusted such that experimental findings about the grain structures in solder joints are reproduced to some extent. Therefore, to address the variety of possible grain structures and its effect on the damage process, engineers and researchers are now able to generate a series of grain structures and perform crack growth calculations in parallel. The fact that this procedure can be automated completely is crucial for the industrial application. To fully capture the failure probability, calculations for a large number of grain structures must be done. Problems during mesh generation would prevent the methodology from becoming an industrial standard: Even if meshing problems are encountered only for a small fraction of the randomly generated grain structures, much user effort would be necessary to correct the problem due to the large overall number of grain structures.

The numerically determined enrichment functions, although they do not represent the weak singularities in the case of creep correctly, still play an important role. They allow us to represent crack tips and re-entrant corners located in the interior of an element in the ansatz space. Without them, the meshing strategy would not result in acceptable numerical solutions unless a very high number of elements is used.

For a small number of grain structures crack growth calculations have been performed and a comparison with experimental results has been done. The simulation and the experimental results are in good agreement. Crack growth as well as the crack patterns are captured well by the simulation. The predicted mean crack length and the standard deviation can be used to obtain a first estimate about the failure probability.

Further research has to be done to develop this method. The simulation should be implemented for three-dimensional structures. Experimental investigations of solder joints should be performed in order to make more precise statements about recrystallization, grain shapes and other characteristic features of the grain structure. The recrystallization process and the soldering process itself could be included in the simulation.

The procedure is a good candidate to capture phenomena such as statistical outliers due to unfavorable microstructures as mentioned earlier. The basic mechanisms of crack growth in solder joints undergoing thermal cycling are

included in the simulation. Applying this procedure to different assemblies and joint geometries may show that the crack growth constant c_{cr} is independent of a particular joint shape and temperature profile. However, this remains to be tested.

A. Stress-strain relation in cylindrical coordinates

The stress-strain relationship in equation (3.6) will now be derived. The derivation has been reproduced from the work of Li et al. [26]. In this section stresses are generally written in cylindrical coordinates. The stiffness tensor $\boldsymbol{D}$ is also assumed to be given in a cylindrical coordinate system and will be written in Voigt notation. The plane strain equilibrium equations in cylindrical coordinates are given by:

$$\frac{\partial \sigma_{rr}}{\partial r} + \frac{1}{r}\frac{\partial \sigma_{r\theta}}{\partial \theta} + \frac{\sigma_{rr} - \sigma_{\theta\theta}}{r} = 0 \tag{A.1}$$

$$\frac{\partial \sigma_{r\theta}}{\partial r} + \frac{1}{r}\frac{\partial \sigma_{\theta\theta}}{\partial \theta} + \frac{2\sigma_{r\theta}}{r} = 0 \tag{A.2}$$

$$\frac{\partial \sigma_{rz}}{\partial r} + \frac{1}{r}\frac{\partial \sigma_{\theta z}}{\partial \theta} + \frac{\sigma_{rz}}{r} = 0 \tag{A.3}$$

We perform the following variable changes:

$$\begin{aligned} &\xi = \ln(r), \quad r = \exp(\xi), \quad s_{rr} = r\sigma_{rr} \\ &\sigma_{rr} = \frac{s_{rr}}{r}, \quad s_{r\theta} = r\sigma_{r\theta}, \quad \sigma_{r\theta} = \frac{s_{r\theta}}{r}, \quad \text{etc.} \end{aligned} \tag{A.4}$$

Using these variable changes the equilibrium equations can be rewritten as:

$$\frac{\partial s_{r\theta}}{\partial \theta} = s_{\theta\theta} - \frac{\partial s_{rr}}{\partial \xi} \tag{A.5}$$

$$\frac{\partial s_{\theta\theta}}{\partial \theta} = \frac{\partial s_{r\theta}}{\partial \xi} - s_{r\theta} \tag{A.6}$$

$$\frac{\partial s_{\theta z}}{\partial \theta} = -\frac{\partial s_{rz}}{\partial \xi} - s_{r\theta} \tag{A.7}$$

We define the following variable vectors:

$$\boldsymbol{\zeta}_2 := \begin{bmatrix} s_{\theta\theta} \\ s_{r\theta} \\ s_{\theta z} \end{bmatrix}, \quad \hat{\boldsymbol{\zeta}} := \begin{bmatrix} s_{rr} \\ s_{zz} \\ s_{rz} \end{bmatrix} \tag{A.8}$$

Thus, the equilibrium equations can be written as:

$$\frac{\partial \boldsymbol{\zeta}_2}{\partial \theta} = \boldsymbol{E}_1 \boldsymbol{\zeta}_2 + \boldsymbol{E}_2 \frac{\partial \boldsymbol{\zeta}_2}{\partial \xi} + \boldsymbol{E}_3 \frac{\partial \hat{\boldsymbol{\zeta}}}{\partial \xi} \tag{A.9}$$

The matrices $\boldsymbol{E}_1$, $\boldsymbol{E}_2$ and $\boldsymbol{E}_3$ in equation (A.9) are defined by:

$$\boldsymbol{E}_1 := \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{E}_2 := \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{A.10}$$

$$\boldsymbol{E}_3 := \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

If the displacement components are independent of the z-axis, the relations between the strain and the displacement components are:

$$\epsilon_{rr} = \frac{\partial u_r}{\partial r} \tag{A.11}$$

$$\epsilon_{\theta\theta} = \frac{1}{r}\left(u_r + \frac{\partial u_\theta}{\partial \theta}\right) \tag{A.12}$$

$$\epsilon_{zz} = 0 \tag{A.13}$$

$$\epsilon_{r\theta} = \frac{1}{r}\frac{\partial u_r}{\partial \theta} + \frac{\partial u_\theta}{\partial r} - \frac{u_\theta}{r} \tag{A.14}$$

$$\epsilon_{rz} = \frac{\partial u_z}{\partial r} \tag{A.15}$$

$$\epsilon_{\theta z} = \frac{1}{r}\frac{\partial u_z}{\partial \theta} \tag{A.16}$$

Using equation (A.11) and the variable changes (A.4) results in the following stress-strain relationship:

$$\begin{bmatrix} s_{rr} \\ s_{\theta\theta} \\ s_{zz} \\ s_{r\theta} \\ s_{rz} \\ s_{\theta z} \end{bmatrix} = \begin{bmatrix} D_{12} & D_{14} & D_{16} \\ D_{22} & D_{24} & D_{26} \\ D_{32} & D_{34} & D_{36} \\ D_{42} & D_{44} & D_{46} \\ D_{52} & D_{54} & D_{56} \\ D_{62} & D_{64} & D_{66} \end{bmatrix} \begin{bmatrix} \frac{\partial u_\theta}{\partial \theta} \\ \frac{\partial u_r}{\partial \theta} \\ \frac{u_z}{\partial \theta} \end{bmatrix} + \begin{bmatrix} -D_{14} & D_{12} & 0 \\ -D_{24} & D_{22} & 0 \\ -D_{34} & D_{32} & 0 \\ -D_{44} & D_{42} & 0 \\ -D_{54} & D_{52} & 0 \\ -D_{64} & D_{62} & 0 \end{bmatrix} \begin{bmatrix} u_\theta \\ u_r \\ u_z \end{bmatrix} + \begin{bmatrix} D_{14} & D_{11} & D_{15} \\ D_{24} & D_{21} & D_{25} \\ D_{34} & D_{31} & D_{35} \\ D_{44} & D_{41} & D_{45} \\ D_{54} & D_{51} & D_{55} \\ D_{64} & D_{61} & D_{65} \end{bmatrix} \begin{bmatrix} \frac{\partial u_\theta}{\partial \xi} \\ \frac{\partial u_r}{\partial \xi} \\ \frac{u_z}{\partial \xi} \end{bmatrix} \tag{A.17}$$

We define a displacement vector as:

$$\boldsymbol{\zeta}_1 := \begin{bmatrix} u_\theta \\ u_r \\ u_z \end{bmatrix} \tag{A.18}$$

We furthermore define the following matrices:

$$\boldsymbol{C}_d := \begin{bmatrix} D_{22} & D_{24} & D_{62} \\ D_{42} & D_{44} & D_{64} \\ D_{62} & D_{64} & D_{66} \end{bmatrix}, \quad \boldsymbol{C}_{d1} := \begin{bmatrix} D_{12} & D_{14} & D_{16} \\ D_{32} & D_{34} & D_{36} \\ D_{52} & D_{54} & D_{56} \end{bmatrix}, \tag{A.19}$$

$$\boldsymbol{C}_e := \begin{bmatrix} -D_{24} & D_{22} & 0 \\ -D_{44} & D_{42} & 0 \\ -D_{64} & D_{62} & 0 \end{bmatrix}, \quad \boldsymbol{C}_{e1} := \begin{bmatrix} -D_{14} & D_{12} & 0 \\ -D_{34} & D_{32} & 0 \\ -D_{54} & D_{52} & 0 \end{bmatrix},$$

$$\boldsymbol{C}_f := \begin{bmatrix} D_{24} & D_{21} & D_{25} \\ D_{44} & D_{41} & D_{45} \\ D_{64} & D_{61} & D_{65} \end{bmatrix}, \quad \boldsymbol{C}_{f1} := \begin{bmatrix} D_{14} & D_{11} & D_{15} \\ D_{34} & D_{31} & D_{35} \\ D_{54} & D_{51} & D_{55} \end{bmatrix}$$

By using these definitions we can rewrite the stress-strain relationship as:

$$\boldsymbol{\zeta}_2 = \boldsymbol{C}_d \frac{\partial \boldsymbol{\zeta}_1}{\partial \theta} + \boldsymbol{C}_e \boldsymbol{\zeta}_1 + \boldsymbol{C}_f \frac{\partial \boldsymbol{\zeta}_1}{\partial \xi} \tag{A.20}$$

$$\hat{\boldsymbol{\zeta}} = \boldsymbol{C}_{d1} \frac{\partial \boldsymbol{\zeta}_1}{\partial \theta} + \boldsymbol{C}_{e1} \boldsymbol{\zeta}_1 + \boldsymbol{C}_{f1} \frac{\partial \boldsymbol{\zeta}_1}{\partial \xi} \tag{A.21}$$

Equivalently we can write:

$$\frac{\partial \boldsymbol{\zeta}_1}{\partial \theta} = \boldsymbol{C}_d^{-1} \left(\boldsymbol{\zeta}_2 - \boldsymbol{C}_e \boldsymbol{\zeta}_1 - \boldsymbol{C}_f \frac{\boldsymbol{\zeta}_1}{\partial \xi} \right) \tag{A.22}$$

$$\begin{aligned} \hat{\boldsymbol{\zeta}} =& \boldsymbol{C}_{d1} \boldsymbol{C}_d^{-1} \boldsymbol{\zeta}_2 + \left(\boldsymbol{C}_{e1} - \boldsymbol{C}_{d1} \boldsymbol{C}_d^{-1} \boldsymbol{C}_e \right) \boldsymbol{\zeta}_1 \\ &+ \left(\boldsymbol{C}_{f1} - \boldsymbol{C}_{d1} \boldsymbol{C}_d^{-1} \boldsymbol{C}_f \right) \frac{\boldsymbol{\zeta}_1}{\partial \xi} \end{aligned} \tag{A.23}$$

$$= \boldsymbol{C}_{d1} \boldsymbol{C}_d^{-1} \boldsymbol{\zeta}_2 + \left(\boldsymbol{C}_{f1} - \boldsymbol{C}_{d1} \boldsymbol{C}_d^{-1} \boldsymbol{C}_f \right) \frac{\boldsymbol{\zeta}_1}{\partial \xi} \tag{A.24}$$

The relationship $\boldsymbol{C}_{f1} - \boldsymbol{C}_{d1} \boldsymbol{C}_d^{-1} \boldsymbol{C}_f = \boldsymbol{0}$ was used in (A.23). The inversion of the matrix $\boldsymbol{C}_d$ is permitted since the strain energy is always positive.
We define the following differential operators:

$$\boldsymbol{H}_{11} := \boldsymbol{E}_1 - \boldsymbol{C}_d^{-1} \boldsymbol{C}_f \frac{\partial}{\partial \xi} \tag{A.25}$$

$$\boldsymbol{H}_{12} := \boldsymbol{C}_d^{-1} \tag{A.26}$$

$$\boldsymbol{H}_{21} := \boldsymbol{E}_3 \left(\boldsymbol{C}_{f1} - \boldsymbol{C}_{d1} \boldsymbol{C}_d^{-1} \boldsymbol{C}_f \right) \frac{\partial^2}{\partial \xi^2} \tag{A.27}$$

$$\boldsymbol{H}_{22} := \boldsymbol{E}_1 + \left(\boldsymbol{E}_2 + \boldsymbol{E}_3 \boldsymbol{C}_{d1} \boldsymbol{C}_d^{-1} \right) \frac{\partial}{\partial \xi} \tag{A.28}$$

Then we obtain the governing equations for the posed problem as:

$$\frac{\partial \boldsymbol{\zeta}_1}{\partial \theta} = \boldsymbol{H}_{11}\boldsymbol{\zeta}_1 + \boldsymbol{H}_{12}\boldsymbol{\zeta}_2 \tag{A.29}$$

$$\frac{\partial \boldsymbol{\zeta}_2}{\partial \theta} = \boldsymbol{H}_{21}\boldsymbol{\zeta}_1 + \boldsymbol{H}_{22}\boldsymbol{\zeta}_2 \tag{A.30}$$

The relationship $-\boldsymbol{C}_d^{-1}\boldsymbol{C}_e = \boldsymbol{E}_1$ was used here. We define the vector $\boldsymbol{\zeta}$ as:

$$\boldsymbol{\zeta} := \begin{bmatrix} \boldsymbol{\zeta}_1^T & \boldsymbol{\zeta}_2^T \end{bmatrix}^T \tag{A.31}$$

Then the equilibrium equations become:

$$\frac{\partial \boldsymbol{\zeta}}{\partial \theta} = \begin{bmatrix} \boldsymbol{H}_{11} & \boldsymbol{H}_{12} \\ \boldsymbol{H}_{21} & \boldsymbol{H}_{22} \end{bmatrix} \boldsymbol{\zeta} := \boldsymbol{H}\boldsymbol{\zeta} \tag{A.32}$$

Using the assumption (3.5) in equation (A.32) results in equation (3.6) with a matrix $\boldsymbol{H}$ depending only on θ and λ.

B. Woodbury matrix identity

We will now derive the Woodbury matrix identity as it is used in equation (3.79). Multiplying the right-hand side in equation (3.79) from the left with $\boldsymbol{K} + \boldsymbol{N}_P^T \boldsymbol{N}_P$ results in:

$$\begin{aligned}
&\left[\boldsymbol{K} + \boldsymbol{N}_P^T \boldsymbol{N}_P\right] \left[\boldsymbol{K}^{-1} - \boldsymbol{K}^{-1} \boldsymbol{N}_P^T \left[\boldsymbol{I} + \boldsymbol{N}_P \boldsymbol{K}^{-1} \boldsymbol{N}_P^T\right]^{-1} \boldsymbol{N}_P \boldsymbol{K}^{-1}\right] && \text{(B.1)} \\
=&\boldsymbol{I} + \boldsymbol{N}_P^T \boldsymbol{N}_P \boldsymbol{K}^{-1} - \boldsymbol{N}_P^T \left[\boldsymbol{I} + \boldsymbol{N}_P \boldsymbol{K}^{-1} \boldsymbol{N}_P^T\right]^{-1} \boldsymbol{N}_P \boldsymbol{K}^{-1} && \text{(B.2)} \\
&- \boldsymbol{N}_P^T \boldsymbol{N}_P \boldsymbol{K}^{-1} \boldsymbol{N}_P^T \left[\boldsymbol{I} + \boldsymbol{N}_P \boldsymbol{K}^{-1} \boldsymbol{N}_P^T\right]^{-1} \boldsymbol{N}_P \boldsymbol{K}^{-1} \\
=&\boldsymbol{I} + \boldsymbol{N}_P^T \boldsymbol{N}_P \boldsymbol{K}^{-1} && \text{(B.3)} \\
&+ \left[-\boldsymbol{N}_P^T - \boldsymbol{N}_P^T \boldsymbol{N}_P \boldsymbol{K}^{-1} \boldsymbol{N}_P^T\right] \left[\boldsymbol{I} + \boldsymbol{N}_P \boldsymbol{K}^{-1} \boldsymbol{N}_P^T\right]^{-1} \boldsymbol{N}_P \boldsymbol{K}^{-1} \\
=&\boldsymbol{I} + \boldsymbol{N}_P^T \boldsymbol{N}_P \boldsymbol{K}^{-1} && \text{(B.4)} \\
&- \boldsymbol{N}_P^T \left[\boldsymbol{I} + \boldsymbol{N}_P \boldsymbol{K}^{-1} \boldsymbol{N}_P^T\right] \left[\boldsymbol{I} + \boldsymbol{N}_P \boldsymbol{K}^{-1} \boldsymbol{N}_P^T\right]^{-1} \boldsymbol{N}_P \boldsymbol{K}^{-1} \\
=&\boldsymbol{I} + \boldsymbol{N}_P^T \boldsymbol{N}_P \boldsymbol{K}^{-1} - \boldsymbol{N}_P^T \boldsymbol{N}_P \boldsymbol{K}^{-1} && \text{(B.5)} \\
=&\boldsymbol{I} && \text{(B.6)}
\end{aligned}$$

Since the left-hand side of equation (3.79) also becomes the identity if multiplied from the left with $\boldsymbol{K} + \boldsymbol{N}_P^T \boldsymbol{N}_P$, we can conclude that equation (3.79) is correct.

Bibliography

[1] IS. Liu. *Continuum mechanics*. Springer Verlag, 2002.

[2] D. Braess. *Finite Elemente*. Springer Verlag, 2003.

[3] E. Kreyszig. *Introductory functional analysis with applications*. Wiley, 1978.

[4] HW. Alt. *Lineare Funktionalanalysis: Eine anwendungsorientierte Einführung*. Springer Verlag, 2006.

[5] C. Grossmann, HG. Roos, and M. Stynes. *Numerical treatment of partial differential equations*. Springer Verlag, 2007.

[6] JE. Marsden and Hughes TJR. *Mathematical foundations of elasticity*. Dover publications, 1994.

[7] JM. Melenk and I. Babuška. The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139(1-4):289–314, 1996.

[8] T. Belytschko and T. Black. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering*, 45:601–620, 1999.

[9] GN. Wells and LJ. Sluys. A new method for modelling cohesive cracks using finite elements. *International Journal for Numerical Methods in Engineering*, 50(12):2667–2682, 2001.

[10] E. Béchet, H. Minnebo, N. Moës, and B. Burgardt. Improved implementation and robustness study of the X-FEM for stress analysis

around cracks. *International Journal for Numerical Methods in Engineering*, 64(8):1033–1056, 2005.

[11] N. Sukumar, DL. Chopp, N. Moës, and T. Belytschko. Modeling holes and inclusions by level sets in the extended finite element method. *International Journal for Numerical Methods in Engineering*, 190(47):6183–6200, 2001.

[12] TP. Fries. A corrected X-FEM approximation without problems in blending elements. *International Journal for Numerical Methods in Engineering*, 75(5):503–532, 2008.

[13] N. Sukumar, DJ. Srolovitz, TJ. Baker, and JH. Prévost. Brittle fracture in polycrystalline microstructures with the extended finite element method. *International Journal for Numerical Methods in Engineering*, 56(14):2015–2037, 2003.

[14] N. Moës, M. Cloirec, P. Cartraud, and JF. Remacle. A computational approach to handle complex microstructure geometries. *Computer Methods in Applied Mechanics and Engineering*, 192(28-30):3163–3177, 2003.

[15] A. Simone, CA. Duarte, and E. van der Giessen. A generalized finite element method for polycrystals with discontinuous grain boundaries. *International Journal for Numerical Methods in Engineering*, 67:1122–1145, 2006.

[16] N. Moës, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46(1):131–150, 1999.

[17] C. Daux, N. Moës, J. Dolbow, N. Sukumar, and T. Belytschko. Arbitrary branched and intersecting cracks with the extended finite element method. *International Journal for Numerical Methods in Engineering*, 48:1741–1760, 2000.

[18] SC. Eisenstat, HC. Elman, and MH. Schultz. Variational iterative methods for nonsymmetric systems of linear equations. *SIAM Journal for Numerical Analysis*, 20(2), 1983.

[19] S. Natarajan, SPA. Bordas, and DR. Mahapatra. On numerical integration of discontinuous approximations in partition of unity finite elements. *IUTAM Symposium on Multi-Functional Material Structures and Systems*, pages 297–304, 2010.

[20] HR. Schwarz and N. Köckler. *Numerische Mathematik*. Teubner Verlag, 2006.

[21] J. Stoer and R. Bulirsch. *Introduction to numerical analysis*. Springer Verlag, 2002.

[22] T. Belytschko, N. Moës, S. Usui, and C. Parimi. Arbitrary discontinuities in finite elements. *International Journal for Numerical Methods in Engineering*, 50(4):993–1013, 2001.

[23] A. Menk and SPA. Bordas. Numerically determined enrichment functions for the extended finite element method and applications to bimaterial anisotropic fracture and polycrystals. *International Journal for Numerical Methods in Engineering*, 83(7):805–828, 2010.

[24] E. Chahine, P. Laborde, and Y. Renard. Spider-X-FEM, an extended finite element variant for partially unknown crack-tip displacement. *European Journal of Computational Mechanics*, 15(5-7):625–636, 2008.

[25] H. Waisman and T. Belytschko. Parametric enrichment adaptivity by the extended finite element method. *International Journal for Numerical Methods in Engineering*, 73(12):1671–1692, 2008.

[26] J. Li, XB. Zhang, and N. Recho. Stress singularities near the tip of a two-dimensional notch formed from several elastic anisotropic materials. *International Journal of Fracture*, 107(4):379–395, 2001.

[27] JCW. Vroonhoven. Stress singularities in bi-material wedges with adhesion and delamination. *Fatigue & Fracture of Engineering Materials & Structures*, 15(2):159–171, 1992.

[28] A. Menk and SPA. Bordas. A robust preconditioning technique for the extended finite element method. *International Journal for Numerical Methods in Engineering*, (DOI:10.1002/nme.3032).

[29] C. Farhat and F. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *International Journal for Numerical Methods in Engineering*, 32(6):1205–1227, 1991.

[30] E. Wyart, M. Duflot, D. Coulon, P. Martiny, T. Pardoen, JF. Remacle, and F. Lani. Substructuring FE–XFE approaches applied to three-dimensional crack propagation. *Journal of Computational and Applied Mathematics*, 215(2):626–638, 2008.

[31] P. Kerfriden, O. Allix, and P. Gosselet. A three-scale domain decomposition method for the 3D analysis of debonding in laminates. *Computational Mechanics*, 44(3):343–362, 2009.

[32] P. Gosselet and C. Rey. Non-overlapping domain decomposition methods in structural mechanics. *Archives of Computational Methods in Engineering*, 13(4):515–572, 2006.

[33] E. Budyn, G. Zi, N. Moës, and T. Belytschko. A method for multiple crack growth in brittle materials without remeshing. *International Journal for Numerical Methods in Engineering*, 61:1741–1770, 2004.

[34] GH. Golub and CF. Van Loan. *Matrix computations*. Johns Hopkins University Press, 1996.

[35] A. Menk and SPA. Bordas. Crack growth calculations in solder joints based on microstructural phenomena with X-FEM. *Computational Materials Science*, 50(3):1145–1156, 2011.

[36] CA. Duarte, LG. Reno, and A. Simone. A high-order generalized FEM for through-the-thickness branched cracks. *International Journal for Numerical Methods in Engineering*, 72(3):325–351, 2007.

[37] SPA. Bordas, M. Duflot, and P. Le. A simple error estimator for extended finite elements. *Communications in Numerical Methods in Engineering*, 24(11):961–971, 2008.

[38] R. Ribeaucourt, MC. Baietto-Dubourg, and A. Gravouil. A new fatigue frictional contact crack propagation model with the coupled X-FEM/LATIN method. *Computer Methods in Applied Mechanics and Engineering*, 196(33-34):3230–3247, 2007.

[39] J. Dolbow, N. Moës, and T. Belytschko. An extended finite element method for modeling crack growth with frictional contact. *Computer Methods in Applied Mechanics and Engineering*, 190(51-52):6825–6846, 2001.

[40] OC. Zienkiewicz and IC. Cormeau. Visco-plasticity and creep in elastic solids - a unified numerical solution approach. *International Journal for Numerical Methods in Engineering*, 8(4):821–845, 1974.

[41] T. Menouillard, J. Réthoré, A. Combescure, and H. Bung. Efficient explicit time stepping for the extended finite element method (X-FEM). *International Journal for Numerical Methods in Engineering*, 68(9):911–939, 2006.

[42] T. Elguedj, A. Gravouil, and A. Combescure. Appropriate extended functions for X-FEM simulation of plastic fracture mechanics. *Computer Methods in Applied Mechanics and Engineering*, 195(7-8):501–515, 2006.

[43] J. Liang, Z. Zhang, JH. Prevost, and Z. Suo. Time-dependent crack behavior in an integrated structure. *International Journal of Fracture*, 125(3):335–348, 2004.

[44] S. Wiese, E. Meusel, and KJ. Wolter. Microstructural dependence of constitutive properties of eutectic SnAg and SnAgCu solders. *Electronic Components and Technology Conference, 2003. Proceedings. 53rd*, pages 197–206, 2003.

[45] Q. Zhang, A. Dasgupta, and P. Haswell. Viscoplastic constitutive properties and energy-partitioning model of lead-free Sn3.9Ag0.6Cu solder alloy. *Electronic Components and Technology Conference, 2003. Proceedings. 53rd*, pages 1862–1868, 2003.

[46] S. Deplanque. *Lifetime prediction for solder die-attach in power applications by means of primary and secondary creep.* PhD thesis, TU Cottbus, 2007.

[47] DR. Shirley, HR. Ghorbani, and JK. Spelt. Effect of primary creep

and plasticity in the modeling of thermal fatigue of SnPb and SnAgCu solder joints. *Microelectronics Reliability*, 48(3):455–470, 2008.

[48] A. Schubert, R. Dudek, E. Auerswald, A. Gollbardt, B. Michel, and H. Reichl. Fatigue life models for SnAgCu and SnPb solder joints evaluated by experiments and simulation. *Electronic Components and Technology Conference, 2003. Proceedings. 53rd*, pages 603–610, 2003.

[49] FX. Che and JHL. Pang. Thermal fatigue reliability analysis for PBGA with Sn-3.8 Ag-0.7 Cu solder joints. *Electronics Packaging Technology Conference, 2004. EPTC 2004. Proceedings of 6th*, pages 787–792, 2004.

[50] A. Syed. Accumulated creep strain and energy density based thermal fatigue life prediction models for SnAgCu solder joints. *Electronic Components and Technology Conference, 2004. Proceedings. 54th*, pages 737–746, 2004.

[51] A. Schubert, R. Dudek, H. Walter, E. Jung, A. Gollhardt, B. Michel, and H. Reichl. Reliability assessment of flip-chip assemblies with lead-free solder joints. *Electronic Components and Technology Conference, 2002. Proceedings. 52nd*, pages 1246–1255, 2002.

[52] R. Darveaux. Effect of simulation methodology on solder joint crack growth correlation. *Electronic Components and Technology Conference, 2000. Proceedings of the 50th*, pages 1048–1058, 2000.

[53] MA. Matin, JGA. Theeven, WP. Vellinga, and MGD. Geers. Correlation between localized strain and damage in shear-loaded Pb-free solders. *Microelectronics and Reliability*, 47(8):1262–1272, 2007.

[54] O. Fouassier, JM. Heintz, J. Chazelas, PM. Geffroy, and JF. Silvain. Microstructural evolution and mechanical properties of SnAgCu alloys. *Journal of Applied Physics*, 100:043519, 2006.

[55] M. Erinc, PJG. Schreurs, GQ. Zhang, and MGD. Geers. Characterization and fatigue damage simulation in SAC solder joints. *Microelectronics and Reliability*, 44(9-11):1287–1292, 2004.

[56] MA. Matin, WP. Vellinga, and MGD. Geers. Thermomechanical fatigue damage evolution in SAC solder joints. *Materials Science & Engineering A*, 445:73–85, 2007.

[57] M. Erinc, PJG. Schreurs, GQ. Zhang, and MGD. Geers. Microstructural damage analysis of SnAgCu solder joints and an assessment on indentation procedures. *Journal of Materials Science: Materials in Electronics*, 16(10):693–700, 2005.

[58] S. Wiese, E. Meusel, and KJ. Wolter. Microstructural dependence of constitutive properties of eutectic SnAg and SnAgCu solders. *Electronic Components and Technology Conference, 2003. Proceedings. 53rd*, pages 197–206, 2003.

[59] S. Choi, KN. Subramanian, JP. Lucas, and TR. Bieler. Thermomechanical fatigue behavior of SnAg solder joints. *Journal of Electronic Materials*, 29(10):1249–1257, 2000.

[60] HY. Lu, H. Balkan, and KY. Simon. Microstructure evolution of the SnAgCu interconnect. *Microelectronics Reliability*, 46(7):1058–1070, 2006.

[61] AU. Telang, TR. Bieler, A. Zamiri, and F. Pourboghrat. Incremental recrystallization/grain growth driven by elastic strain energy release in a thermomechanically fatigued lead-free solder joint. *Acta Materialia*, 55(7):2265–2277, 2007.

[62] TR. Bieler, H. Jiang, LP. Lehman, T. Kirkpatrick, EJ. Cotts, and B. Nandagopal. Influence of Sn grain size and orientation on the thermomechanical response and reliability of Pb-free solder joints. *Components and Packaging Technologies, IEEE Transactions on*, 31(2):370–381, 2008.

[63] LP. Lehman, SN. Athavale, TZ. Fullem, AC. Giamis, RK. Kinyanjui, M. Lowenstein, K. Mather, R. Patel, D. Rae, and J. Wang. Growth of Sn and intermetallic compounds in Sn-Ag-Cu solder. *Journal of Electronic Materials*, 33(12):1429–1439, 2004.

[64] B. Vandevelde, M. Gonzalez, P. Limaye, P. Ratchev, and E. Beyne. Thermal cycling reliability of SnAgCu and SnPb solder joints: a comparison for several IC-packages. *Microelectronics Reliability*, 47(2-3):259–265, 2007.

[65] JJ. Sundelin, ST. Nurmi, and TK. Lepistö. Recrystallization behaviour of SnAgCu solder joints. *Materials Science and Engineering: A*, 474(1-2):201–207, 2008.

[66] P. Limaye, B. Vandevelde, D. Vandepitte, and B. Verlinden. Crack growth rate measurement and analysis for WLCSP SnAgCu solder joints. *Circuits Assembly*, 17(2):68, 2006.

[67] DW. Henderson, JJ. Woods, TA. Gosselin, J. Bartelo, DE. King, TM. Korhonen, MA. Korhonen, LP. Lehman, EJ. Cotts, and SK. Kang. The microstructure of Sn in near-eutectic Sn-Ag-Cu alloy solder joints and its role in thermomechanical fatigue. *Journal of Materials Research*, 19(6):1608–1612, 2004.

[68] LP. Lehman, RK. Kinyanjui, J. Wang, Y. Xing, L. Zavalij, P. Borgesen, and EJ. Cotts. Microstructure and damage evolution in Sn-Ag-Cu solder joints. In *Electronic Components and Technology Conference, 2005. Proceedings of the 55th*, pages 674–681. IEEE, 2005.

[69] J. Gong, C. Liu, PP. Conway, and VV. Silberschmidt. Micromechanical modelling of SnAgCu solder joint under cyclic loading: Effect of grain orientation. *Computational Materials Science*, 39(1):187–197, 2007.

[70] M. Erinc, PJG. Schreurs, and MGD. Geers. Intergranular thermal fatigue damage evolution in SnAgCu lead-free solder. *Mechanics of Materials*, 40(10):780–791, 2008.

[71] A. Menk and SPA. Bordas. Influence of the microstructure on the stress state of solder joints during thermal cycling. In *Thermal, Mechanical and Multi-Physics simulation and Experiments in Microelectronics and Microsystems, EuroSimE, Proceedings of the 10th International Conference on*, pages 1–5. IEEE, 2009.

[72] HB. Ghavifekr and B. Michel. Generalized fracture mechanical integral concept and its application in microelectronic packaging technology. *Sensors & Actuators*, 99(1-2):183–187, 2002.

[73] P. Towashiraporn, G. Subbarayan, and CS. Desai. A hybrid model for computationally efficient fatigue fracture simulations at microelectronic assembly interfaces. *International Journal of Solids and Structures*, 42(15):4468–4483, 2005.

[74] KR. Tunga. *Study of tin-silver-copper alloy reliability through material microstructure evolution and laser moire interferometry*. PhD thesis, Georgia Institute of Technology, 2008.

[75] J. Andersson. The influence of grain size variation on metal fatigue. *International Journal of Fatigue*, 27(8):847–852, 2005.

[76] O. Watanabe, HM. Zbib, and E. Takenouchi. Crystal plasticity: Micro-shear banding in polycrystals using Voronoi tessellation. *International Journal of Plasticity*, 14(8):771–788, 1998.

[77] M. Nygårds and P. Gudmundson. Three-dimensional periodic Voronoi grain models and micromechanical FE-simulations of a two-phase steel. *Computational Materials Science*, 24(4):513–519, 2002.

[78] J. Rösler, H. Harders, and M. Bäker. *Mechanisches Verhalten der Werkstoffe*. Teubner Verlag, 2006.

[79] AR. Fix, GA. Lopez, I. Brauer, W. Nüchter, and EJ. Mittemeijer. Microstructural development of Sn-Ag-Cu solder joints. *Journal of Electronic Materials*, 34(2):137–142, 2005.

[80] WP. Mason and HE. Bommel. Dislocation mobility in Sn crystals. *Journal of the Acoustical Society of America*, 28:930–941, 1956.

[81] F. Yang and JCM. Li. Deformation behavior of tin and some tin alloys. *Journal of Materials Science: Materials in Electronics*, 18(1-3):191–210, 2007.