Montanuniversität Leoben

Numerical analysis of a multivariant martensitic phase transformation in nanostructured NiTi considering elastic anisotropy

Diplomarbeit

von

Manuel Petersmann

Leoben, April 2014

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfsmittel bedient habe.

Affidavit

I declare in lieu of oath, that I wrote this thesis and performed the associated research myself, using only literature cited in this volume.

Manuel Petersmann

Leoben, September 2013

Danksagung

Besonderer Dank gilt Univ.-Prof. Dipl.-Ing. Dr.mont. Thomas Antretter, der mir die Gelegenheit gab am Institut für Mechanik zu arbeiten, jederzeit für Fragen und Diskussionen ein offenes Ohr hatte, und der mit vielen Anregungen im Umgang mit ABAQUS und Linux zum Gelingen dieser Arbeit beigetragen hat.

Dank gebührt insbesondere auch Dipl.-Ing. Dr.mont Michael Fischlschweiger, der es mir im Rahmen seiner Dissertation als erstes ermöglichte, für Ihn als studentischer Mitarbeiter am Institut für Mechanik zu arbeiten, und der mich für Modellierung und Simulation begeisterte.

Ich möchte auf diesem Wege auch den viel zu früh verstorbenen Ao. Univ.-Prof. Dipl.-Ing. Dr. phil. Eduard R. Oberaigner für seine Spezialvorlesungen danken, die mich immer wieder inspiriert und motiviert haben.

Ferner danke ich allen Mitarbeitern des Institutes für Mechanik für das gute Arbeitsklima.

Auch meinen Freunden, die ich während meiner Studienzeit kennengelernt habe und die mit mir diesen Weg gegangen sind, sei dafür gedankt.

Schlussendlich möchte ich mich bei meiner ganzen Familie bedanken, die mich während meiner Studienzeit unterstützt hat, vor allem aber bei meinem Vater und meiner Mutter Walter und Edith Petersmann.

Die vorliegende Arbeit wurde im Rahmen des COMET Projektes A1.5 WP4A1.5 WP4 in Zusammenarbeit mit dem Materials Center Leoben und dem Institut für Mechanik der Montanuniversität Leoben erstellt.

Der österreichischen Bundesregierung (insbesondere dem Bundesministerium für Verkehr, Innovation und Technologie und dem Bundesministerium für Wirtschaft, Familie und Jugend) vertreten durch die Österreichische Forschungsförderungsgesellschaft mbH (FFG), und den Ländern Steiermark und Tirol, vertreten durch die Steirische Wirtschaftsförderungsgesellschaft mbH (SFG) sowie die Standortagentur Tirol wird für die finanzielle Unterstützung der Forschungsarbeiten im Rahmen des von der Materials Center Leoben Forschung GmbH abgewickelten K2 Zentrums für „Materials, Processing und Product Engineering“ im Rahmen des Österreichischen COMET Kompetenzzentren Programms sehr herzlich gedankt.

Kurzfassung

NiTi Legierungen sind wegen ihrer vielfältigen mechanischen und funktionellen Eigenschaften aus ingenieurstechnischen, physikalischen bzw medinzinischen Gründen die am häufigsten verwendeten Formgedächtnislegierungen. Der Formgedächtniseffekt beruht auf einer völlig reversiblen martensitischen Phasenumwandlung, die von speziellen transformationsbedingten Steifigkeitsveränderungen und Dehnungsentwicklungen begleitet ist, weshalb sich NiTi Legierungen vor allem für den Einsatz als Sensoren, Aktuatoren, Dämpfungselemente und in der Medizintechnik besonders eignen. Die nachfolgende Arbeit behandelt die Modellierung dieser Transformation in nano- und polykristallinem NiTi, welches beim Abkühlen aus dem Austenitbereich (Hochtemperaturphase) anders als grobkörniges NiTi über eine orthorombische Zwischenphase ab ca. 30°C von einer kubisch geordneten Kristallstruktur höchster Symmetrie in die monokline, niedriger-symmetrische Phase des Martensits umwandelt. Die Transformation erstreckt sich über ein sehr breites Temperaturintervall und ist in einer freien Energiebetrachtung überwiegend thermisch aktiviert. Nach der Definition des monoklinen Kristallgitters würde die Umwandlung von Austenitbereichen im Inneren der kubischen Matrix hohe Spannungen hervorrufen, die energetisch betrachtet sehr ungünstig wären, weshalb sich eine Martensitmorphologie ausbildet, welche die entstehende Verzerrungsenergie minimiert, die sogenannten Zwillinge. Mit Hilfe der nichtlinearen Theorie des Martensits wurde, ausgehend von den Gitterabständen der kubischen bzw. der monoklinen Phase, der Deformationsgradient, der die Umwandlung mathematisch beschreibt, sowie alle möglichen Martensitvarianten und daraus folgende mögliche Zwillingskonfigurationen und deren Deformationen berechnet. Es wurde eine Korngrößenverteilung unter 100nm Durchschnittsdurchmesser angenommen, da hier die Martensitausbildung als vollständig transformiertes Nanokorn, durchzogen aus sich abwechselnden Martensitvarianten, beobachtet wurde. Das Materialverhalten wurde völlig anisotrop mit elastischen Konstanten aus ab initio Rechnungen beschrieben. In diesem Modell zeigt sich, dass die elastische Verzerrungsenergie als Folge der Transformation den größten Energiebeitrag zur Transformationsbarriere in nanokristallinem NiTi darstellt. Ein inkrementeller Algorithmus für die Transformation wurde entwickelt, der in jedem Umwandlungsschritt die absolute Energie minimiert und folglich auf ein absolutes Minimum führen soll. Auf diese Art und Weise erhält man eine Umwandlungskinetik, die mit der von NiTi aus Experimenten bekannten Kinetik sehr gut übereinstimmt.

Abstract

NiTi alloys exhibit diverse mechanical as well as functional properties and are the most commonly used SMAs (shape memory alloys) for engineering as well as medical applications. NiTi's shape memory effect is caused by a fully reversible martensitic transformation. The transformation is accompanied by macroscopic changes in the material's stiffness and strain evolution which makes NiTi particularly relevant for sensors, actuators and damping elements. The following work deals with the modeling of the cubic to monoclinic transformation in nano- and polycrystalline NiTi, which is triggered by temperature. It proceeds in a broad temperature interval starting after the very small temperature interval of intermediate transformation to an orthorombic phase at around 30°C. The high-temperature, high-symmetry phase called austenite is a cubic, ordered crystal and the low-temperature, lower symmetry phase called martensite is monoclinic. In order to accommodate the new phase a twinned crystal structure is formed. Using the nonlinear theory of martensitic transformations, starting only from lattice parameters of the cubic and monoclinic phase, the deformation gradients describing the shape changes of all possible martensitic variants and variant-pairs forming a twin are calculated. A nano-grain distribution below 100nm average diameter is modeled, since for these small grain sizes the preferred martensite morphology turns out to consist of a single laminate of alternating variants. The material itself is modeled as a thermoelastic solid. Anisotropic material behavior with elastic constants from ab initio calculations are used in combination with locally random orientations at the grain level. It was found that the elastic strain energy constitutes the main contribution to the total energy barrier. In this work an incremental algorithm for the transformation was developed based on an energy minimizing principle. The so obtained transformation kinetics agrees with experimental evidence reported in the literature.

Contents

1. Introduction and Motivation

In this work martensitic transformation is described using a thermodynamic approach by calculating the free-energies of the different phases as this is frequently done in computational materials science. However, this model is furthermore combined with a geometrically based approach on the atomic level outlined in chapter 2. The extension of a mere thermodynamic approach seems reasonable since a martensitic transformation occurs by lattice distortion, which makes it important to investigate these transformations with respect to lattice dynamic properties rather than purely thermodynamic ones.

1.1. Martensitic Phase Transformations

1.1.1. Classification and Terminology of Martensitic Transformations

Martensitic transformations are heterogeneous, athermal, diffusionless, structural solid to solid phase transformations. This section is intended to clarify the terminology relevant to martensitic transformations. "Heterogeneous" means that at each intermediate stage of the transformation the assembly can be divided microscopically into distinct regions, some have transformed while others have not. "Athermal" means that although the transformation is triggered by a temperature dependent chemical potential, the transformation itself occurs in such a small time increment that it cannot be associated with a change of heat or temperature. The term "diffusionless" already suggests that the transformation does not require long-range diffusion during the phase change, but only small atomic movements over usually less than the interatomic distances. A general classification of diffusionless phase transformations is shown in Figure 1.1. Since no atomic migration is involved these transformations progress in a time independent fashion, with the motion of the interface between the two phases being limited only by the speed of sound.

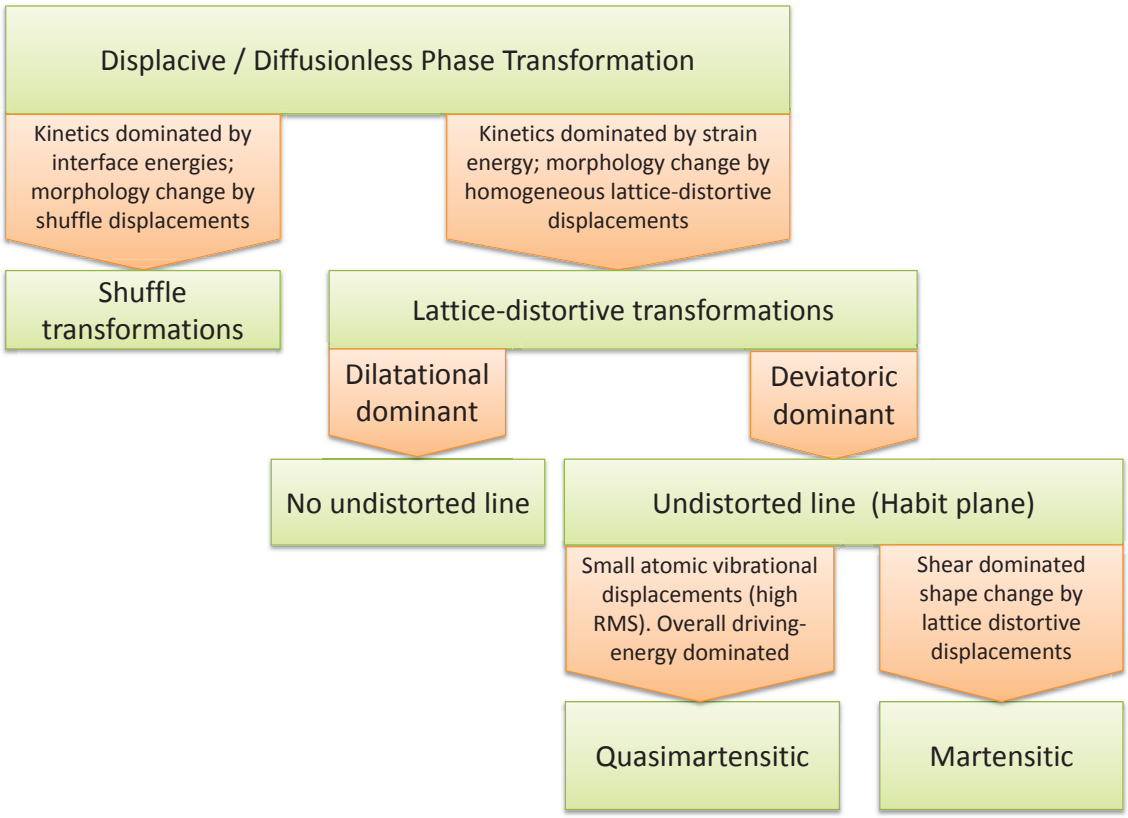

Figure 1.1.: Classification scheme for diffusionless phase transformations as proposed by Cohen et al. [1]

A martensitic phase transformation occurs as a quick regional change of the crystal lattice, while the chemical composition remains constant. In general, a critical cooling rate has to be applied to the material in order to inhibit diffusion and receive a martensitic transformation. Martensitic phase transformations have been known since the early years of materials science. The first observed case and most popular example is the martensitic transformation in carbon alloyed iron, i.e. steel, where austenite, irons face-centered cubic (fcc) phase with evenly distributed carbon, transforms to a carbon supersaturated body-centered cubic (bcc) lattice phase called martensite. Since its first classification, over the years this kind of transformation has also been observed in non-ferrous alloys, where it revealed a variety of fascinating phenomena, including the effects of thermoelasticity, superelasticity and the shape-memory effect. In non-ferrous materials the martensite may exhibit quite different features compared to ferrous materials. A qualitative comparison between ferrous and non-ferrous martensites is presented in Table 1.1. However, martensitic transformations are not solely restricted to metallic alloys, they are also found in polymers, ceramics and even in proteins (e.g. the Bacteriophage T-4).

Table 1.1.: A qualitative comparison between ferrous and non-ferrous martensites according to Delaey et al [2]

Property	Ferrous martensite	Non-ferrous martensite
Nature of alloying	Interstitial and/or substitutional	Substitutional
Hardness	Martensite is much harder than austenite	Martensite is only slightly harder and can even be softer than austenite
Transformation hysteresis	Large	Small to very small
Transformation strain	Relatively large	Relatively small
Elastic constants of the parent phase	High values near M_s	Low values near M_s
Temperature coefficient of elastic shear constant	Negative near M_s in most cases	Positive near M_s in many cases
Transformation enthalpy	High	Low
Transformation entropy	Large	Small
Chemical driving force	Large	Small
Growth character	Self-accomodation is not obvious	Well developed self-accommodating variants
Kinetics	High rate, „burst“, athermal and/or isothermal transformation	Slower rate, no „burst“, only athermal transformation, thermoelastic balance
Growth front	No single interface transformation observed	Single interface possible
Interface mobility	Low and reversible	High and reversible
Damping capacity of martensite	Low	High

The classification of martensitic transformations is sometimes controversially discussed. It has been pointed out several times in the literature, that an exact border categorizing the nature of martensitic transformations is hard to draw. Nevertheless, much factual material on the morphology, thermodynamics and kinetics of this special type of transformation has accumulated to date. An outlining, yet incomplete list of features according to Roitburd and Kurdjumov [3] is nevertheless given subsequently, as it is considered an adequate guideline.

1. Martensite grows to a certain extent as flat plates, but owing to the high elastic stresses that are building up it consequently forms in lenticular shapes which narrow towards their ends. This is especially true for ferrous martensite as is illustrated in Figure 1.2.

2. A general feature of martensite is the so called “Habit plane”, a crystallographic plane or system of planes along which certain phenomena such as twinning occur. However there are three different interpretations of a martensite habit plane in use [4]

as can be seen in Figure 1.2. The first one is the plane of the plate shaped martensite crystal. Second, in the case of twinned martensite and grain-sizes significantly larger than about 100nm, smooth planar faces also called midribs or junction planes as it is called in so called Herringbone structures shown in Figure 1.7 b) are referred to as habit plane. And third, rarely also the plane boundary of a plate shaped product is referred to as habit plane. In every case the habit plane has a discrete orientation to the crystalline axes of the initial as well as the final phases.

3. Normally there exists a definite orientation relationship between the lattices of the initial and final phases, as is elaborated in more detail in subsection 2.4.3.

4. Transformation changes the shape of the transformed region. This macrodeformation is homogeneous and can be seen as a combination of simple shear along the plane of the plate and dilation or compression normal to that plane.

5. Martensitic crystals have a regular internal structure. For instance, fully transformed nano-grains consist of a sequence of alternating twin variants explained in more detail in subsection 2.5.1.

6. Martensite in general has the tendency towards forming a somehow ordered distribution of plates, which is indicative for the aforementioned definite lattice relations between the phases.

7. Transformation starts only at some deviation from the system's free energy equilibrium, which is mainly depending on temperature, but also on stress.

8. The phase fraction of the transformed phase increases only with increasing deviation from equilibrium conditions. For reversible martensitic transformations this means it can also decrease in the case of free energy approaching back to equilibrium again. For irreversible transformations only a monotonically increasing free energy can cause further transformation.

9. In the formation of martensitic crystals a high growth rate is observed ($10^4 - 10^5$ cms^{-1}), not showing any noticeable temperature dependence, provided other variables such as grain-size are held constant.

10. According to the nucleation kinetics, martensitic transformations are described as athermal and isothermal. For athermal transformations the nucleation rate is high and does not show any temperature dependence. The transformation rapidly reaches a relatively stable state, at which it is necessary to increase the deviation from equilibrium significantly in order to continuously drive the transformation. However

in some reactions there is also a small amount of isothermal transformation to the martensitic phase due to small diffusive contributions.

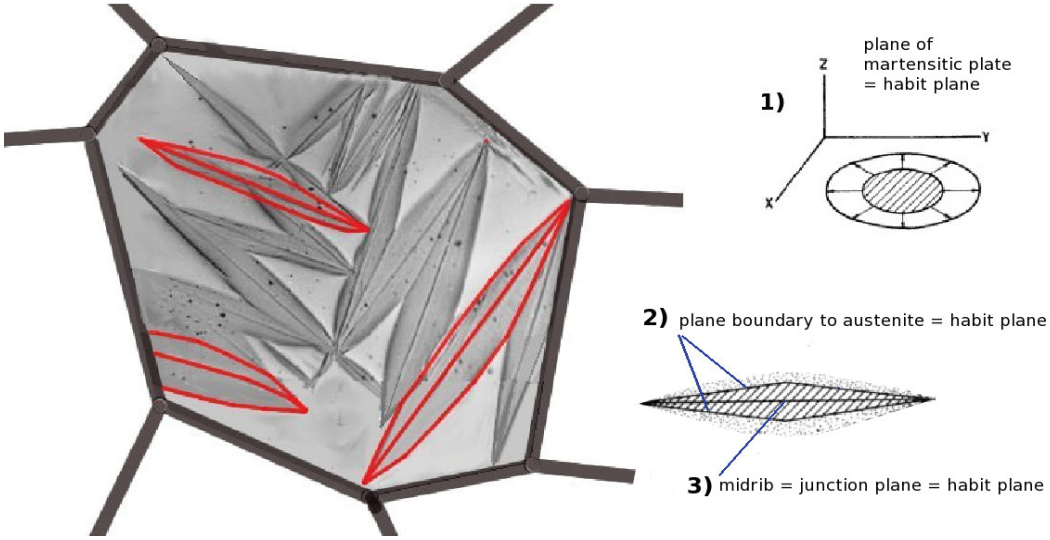

Figure 1.2.: lenticular martensite and the three interpretations of a habit plane

As mentioned above this list combines aspects of thermodynamics, crystallography and kinetics. In the following these aspects shall be reviewed individually.

Chronologically, the first viable theoretical description of martensitic transformations took advantage of the kinetic similarity to plastic deformation. Its theoretical basis was presented by Zener [5]. Here the martensitic transformation is said to occur due to a loss of the mechanical stability of the austenite. Although this hypothesis could later on be experimentally confirmed for special crystal systems, it still remains unclear whether a mechanical instability is necessary for the transformation.

A combined thermodynamic-kinetic approach based on the framework of classical nucleation and growth theory adopted for martensitic transformations was made by the soviet school of physical metallurgy. Kurdjumov as one of the first classified the transformation as a first-order phase transition according to Paul Ehrenfest (student of Ludwig Boltzmann) that proceeds under conditions where the initial phase maintains meta-stability. The definition of the order of a phase transition shall be given here. In principle a system is described by a thermodynamic potential generally referred to as free energy and several forms of the free energy may be formulated based on system criteria. Which form is suitable depends on which thermodynamic variable is held constant within the described process. The two most common forms are the so called Gibbs free energy and the Helmholtz potential. Generally the phase with the lowest free energy will be stable. The thermodynamics of an alloy showing martensitic transformation resembles that of a single-component system [6]. Figure 1.3 shows a free energy over temperature diagram of martensite and austenite. The free energies of two coexisting phases at the transformation temperature and stress state are equal, but their first derivatives with respect

to their thermodynamic variables may not be. If the first derivatives are different, the difference in entropy ΔS, volume ΔV and enthalpy ΔH (in this case the latent heat) become $\neq 0$ between the two phases at the point of transition. Then the transition is known as first order transition as schematically shown in Figure 1.3. Common examples of first-order transitions are all solid - liquid - gas transitions because they involve a discontinuous change in density, being the first derivative of the free energy with respect to the chemical potential. Principally the order of a phase transition can be defined as lowest order of the derivative of the free energy that is discontinuous at the transition state. For a first order phase transition, roughly speaking the free energy curve is a continuous function. Some may consider this method of classification as inaccurate, for it does not

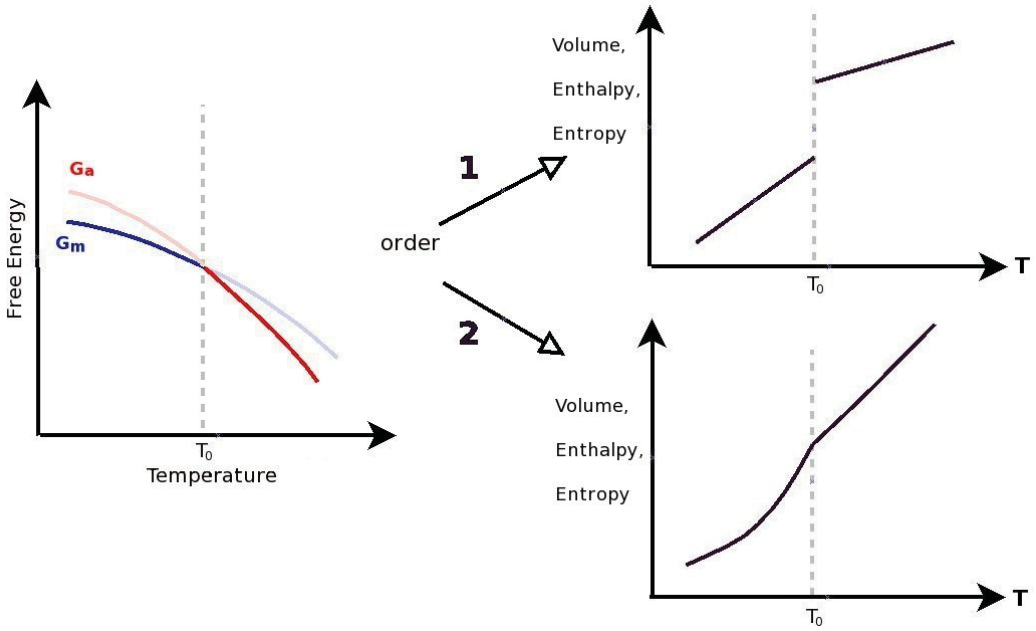

Figure 1.3.: classification of the order of a phase transformation

take into account the case where a derivative of the free energy diverges, i.e. it tends to infinity, e.g. the heat capacity in the case of a ferromagnetic transition. In another definition of order according to the theory of L.D. Landau, phase transitions are described by symmetry-breaking from ordered to unordered phases accompanied by discontinuities of macroscopic properties, called order parameters, such as the deformation of a crystal lattice [7]. It should however be noted that this theory is mainly used for the description of second order phase transitions, where the order parameter continuously approaches zero as the temperature approaches the transformation temperature. An example due to Landau for a symmetry classified transformation in the solid state is the transformation from cubic to tetragonal crystal structure. Although the symmetry changes abruptly upon transformation, one can always determine in which phase a certain domain exists at any given time. Whether the transformation is of first or second order also determines the microstructure: First order transformations form parent/product or heterophase interfaces in addition to product/product interfaces, whereas second order transformations

only form product/product interfaces. In the former case the first plates formed can grow to a larger extent than those formed later, which can lead, e.g. to microstructures with fractal characteristics [8]. Furthermore it is important to restrict this classification to the crystallographic local level, because only there the atoms or molecules, respectively, can change their arrangement immediately within a very small time increment [7]. The relations of symmetries and order parameters between different phases were elaborated extensively in recent years within the framework of statistical physics, experimentally and in simulations. The works inspired by the soviet school of physical metallurgy led to a further distinction of isothermal and thermoelastic martensitic transformations, as well as to the discovery of an intermediate martensite-like transformation resulting in so called Widmanstätten structures [3]. Thermoelasticity is one of the key features characterizing shape-memory alloys (SMAs), where martensitic transformation is fully reversible.

In recent years, experimental as well as theoretical research led to crystallographic theories which describe the formation of martensite due to geometric compatibility of the lattice at an atomic scale. At this scale, neglecting defects, the reconstruction of the lattice causes a mere homogeneous strain deformation of the unit cell, so that the final phase can be regarded as a homogeneously strained initial phase. Mathematically this shape change of the unit cell is described by a deformation gradient (2.13). Such phase transformations, where the states of the phases are completely determined by the strain of the lattice are called strain transformations. At this point the term “displacive” characterizing martensitic transformations is introduced. It denotes a diffusionless, first-order strain transformation at the crystallographic level. In martensitic transformations the local order of the crystal lattice at the emerging interface between the old and new phases is maintained. This fact causes the new and the initial phases’ lattices to share a common plane parallel to their interface. The interfaces therefore have an ordered, so called coherent (or at least semi-coherent) structure. This notion became apparent due to the high growth rate of the martensitic crystals long before transmission electron microscopy enabled to observe and verify these coherent interfaces in situ. The regular coherent structure of the forming interface boundaries thus is responsible for their low energy and high mobility. However they are also the origin of internal stresses that play an essential role for the transformation path of the microstructure and therefore for the transformation kinetics. Residual stresses are self-equilibrated, thus they do not result in a net external force. Furthermore, for a homogeneous deformation it is necessary that the martensite domain is sufficiently mobile, and the slip systems are sufficiently rigid, otherwise plastic deformation would occur instead. The basic idea is that a deformation will best be accommodated by a stress-free, thus strain-energy minimizing microstructure. The material is said to be self-accommodating. Here it is important to keep in mind that not every geometrically possible accommodation will arise, but only those that are strain-energetically most favorable, due to a mutual compensation of the individual strain

fields. If the material is volume preserving the amount of tensile and compressive stresses in it are nearly equal so that a mutual compensation of these stresses by accommodation lead to a barely stressed microstructure. The effectiveness of accommodation moreover depends on the grain morphology, the orientation distribution (possibly textures) and most importantly the number of variants and variant combinations the martensite can form. This latter aspect depends on the lattices of the parent and product phases and their symmetries and is elaborated later in more detail in subsection 2.4.3. However, it should be made clear that not every material that undergoes a martensitic phase transformation can be self-accommodating because some martensitic transformations are not volume preserving, leading to macroscopic stresses being of tensile or compressive nature. The beauty of the crystallographic theories is that they do not contradict other theories, but can rather be seen as an additional constraint for the prediction of the morphological evolution, thus complementing other methods. It describes the formation of martensite based on geometric compatibility reasons necessary for a coherent interface on an atomic level, which generally makes it a structural classification of the phase transition. Among those the martensitic transformation is categorized into a discontinuous (reconstructive) phase transition, where chemical bonds are broken and a continuous one, where they persist. The term reconstructive recently also gained importance considering the reversibility of the transformation. Reversibility means that given appropriate thermodynamic driving forces a single crystal domain of austenite will resume its original lattice structure regardless of the morphology of the (twinned) product phase and regardless of the number of transformation cycles. In the course of the transformation the atoms are displaced less than one lattice parameter each, giving rise to the fact that neighboring atoms at the boundary in the parent lattice remain neighbors in the product lattice after the transformation. This so called “lattice correspondence” suggests that there is no reason why the movement of such an interface should not be reversible. Bhattacharya et al. [9] explained the difference in reversibility with symmetry reasons. They claim a fundamental difference between “weak” and “reconstructive” transformation. In a weak transformation the product and the parent phase must be included in a common finite symmetry group, e.g. the cubic to monoclinic transformation. This means that the group with lower symmetry has to be a subset of the higher symmetry one, as is illustrated in Figure 2.1. By contrast, in reconstructive transformations they are not included in a common finite symmetry group, e.g. cubic to hexagonal. Only weak transformations can occur reversibly, whereas in reconstructive transformations irreversibility is inevitable. Besides the theoretical explanation, this is also observed experimentally for the case of pure iron (Fe) where a reconstructive transformation occurs. However, adding Ni and C for obtaining steel the martensite becomes bct with increasing tetragonality, meaning the transformation becomes weak [9, 10].

The first crystallographic theory was published by Bowles and Mackenzie [11, 12] as well

as Wechsler, Lieberman and Read. It is hence also referred to as phenomenological WLR Theory [13]. Later on, based on ideas of Erickson et al. [14, 15, 16, 17] Ball and James developed an elastically nonlinear theory [18] since large rotations occur in martensitic transformations and if these are linearized then phantom stresses are predicted [19]. Both these theories use the same equations of geometric compatibility for an invariant plane, i.e. an undistorted and unrotated plane with continuous deformations on each side as outlined in section 2.5. The nonlinear theory uses variational formulations of the free energy as a function of deformation and temperature (see (2.18)). The basic assumption of the non linear theory is that the observed microstructures correspond to minimizers or almost minimizers of the strain energy. In both theories operations like deformations and rotations of the parent lattice are represented by matrices. According to the WLR theory the total deformation matrix $\mathbf{P_1}$, also called shape strain, consists of three components. (i) a so called Bain strain $\mathbf{B}$ also denoted as $\mathbf{U}$ as elaborated in subsection 2.4.3, (ii) a lattice invariant shear $\mathbf{P_2}$, i.e. an inhomogeneous shear leaving the martensite structure unchanged, e.g. twinning or slip, and a rotation, here denoted by $\mathbf{R}$ [20] and therefore writes as $\mathbf{P_1} = \mathbf{RP_2B}$. This theory can be used to describe martensitic transformations if the plane and direction of the lattice invariant shear is known, with no limitations on the deformations. In the nonlinear theory, however, deformations are limited by a so called Ericksen-Pitteri neighborhood [21] elaborated in section 2.3. More precisely in the nonlinear theory no reconstructive transformation, e.g. lattice invariant shear is allowed. For certain materials like shape-memory alloys this Ericksen-Pitteri neighborhood suffices to describe transformation deformations. Hence, in SMAs the nonlinear theory is commonly the preferred tool. Ball [19] states that the nonlinear elasticity model incorporates the WLR crystallographic theory of martensite. A remarkable result of the nonlinear theory is that it predicts the formation of very fine microstructures, like those observed in nanostructured materials. Static crystallographic theories are sometimes criticized since the formation of martensite is clearly a pattern formation problem. It should therefore be treated using dynamic models. A historical background to the above mentioned theories can be found in [22] as well as in the introductory remarks of chapter 7 in [21].

Finally another noteworthy aspect is the effect of dislocations, which is generally an upcoming topic in materials science. A so called glissile interface, a term from the theory of dislocations, between parent and product phases is considered necessary in order to obtain a martensitic transformation after all. A glissile dislocation has a Burger's vector that lies in the primary slip plane of the crystal and thus is able to move in that plane. Contrary, a non-glissile or "sessile" dislocation has a Burger's vector that does not lie in the primary slip plane of the crystal making it immobile. A glissile interface can migrate under the action of a suitable driving stress even at very low temperatures, and its movement does not require thermal activation, whereas sessile interfaces require the assistance of thermal fluctuations.

1.1.2. Shape-Memory-Effect (SME)

The term “shape memory” has been used to cover a variety of effects. It is commonly understood as purely thermal shape memory, where no external load is applied during the transformation. Here the shape of the undeformed material at a high temperature is recovered after deforming a specimen at some lower temperature and reheating to the original temperature. Also the effects of pseudo- or superelasticity explained in this section are referred to as shape-memory effect in the case that an external load is present during the transformation. Anyway, the source of the shape-memory effect is always a special case of the martensitic transformation, which is why Wayman [23] also refers to this transformation as “Marmem”, an acronym for martensite memory effect. The transformation in this case is thermoelastic, i.e. fully reversible, as pointed out earlier. Materials that exhibit a purely thermal shape memory upon heating are referred to as showing a one way shape-memory effect (OWSME). Materials that additionally undergo a purely thermally activated change in shape (compared to their undeformed state) upon recooling remember two different shapes: one at low temperatures, and one at the elevated temperatures. This is termed two way shape-memory effect (TWSME). For a material to posses a TWSME it must be trained by multiple transformations under certain conditions as is explained at the end of this section. As an introductory remark suffice it to say that after the training the TWSME entails a repeatable shape change of the material under no applied mechanical load, but only subjected to a cyclic thermal load.

There are two major groups of alloys exhibiting the SME. First, the binary alloys of transition metals. In the periodic table one component is found in the column left and the other one right of the chromium group. The NiTi alloy treated here is a member of this category. The β-phase alloys of noble metals constitute the second class [23]. Most of them have an ordered caesium chloride (CsCl) structure (cubic Bravais lattice) at their high temperature phase and all of them transform martensitically to a low-temperature phase with lower symmetry e.g. monoclinic, orthorombic or tetragonal. The cubic parent phase has the highest possible symmetry of all Bravais lattices, enabling it to form enough variants of the martensite to sufficiently accommodate the stresses arising upon transformation, which is theoretically explained in subsection 2.4.3. The total change in shape of the crystal lattice is mainly a pure shear in an invariant plane strain. A significant change in volume is normally not observed. Therefore a widely known feature of SMAs is that they are volume preserving upon transformation. Besides NiTi Cu-Al-Ni is very popular and has intensively been studied.

The transformation from austenite to martensite is termed forward transformation. Four characteristic temperatures are associated with the transformation, two for the forward and two for the reverse transformation, respectively. They mark the start and finish of

the newly arising phase. In the forward case these temperatures are called martensite start (M_s) and martensite finish (M_f) temperature and for the reverse case austenite start (A_s) and austenite finish (A_f) temperature, respectively. Transformation temperatures of common SMAs like NiTi lie near their operating-temperature which is typically near room temperature. Diffusion does not play a role because critical cooling rates are always exceeded. To understand the peculiarities of the transformation, one must study the transformation concerning temperature and stress, because the magnitude and interaction of these two factors determine the transformation path. As is shown in Figure 1.4 there are four possible transformation paths, which are subsequently described and whose effects are explained.

First the forward transformation without the influence of an external load shall be discussed. Generally, if a domain in the material transforms from one lattice to another internal strains are generated. For domain sizes larger than some atom sizes these strains would cause stresses up to the yield point leading to plastic deformations around the transformed region in order to fit in the surrounding matrix. For plastic deformation to occur dislocations have to be created, increasing the material's internal energy. However, in SMAs twinning is the common mechanism minimizing the overall strain of the deformed domain such that the yield stress is not reached and no plastic deformation accompanied by dislocation creation is necessary. The martensite is said to be self-accommodating. Obviously, the strain energy stored in the elastically distorted lattice upon twinning is lower than the energy to generate dislocations to enable plastic deformation. In the process of twinning martensite forms as a combination of different variants called twins as shown in Figure 1.5. Theoretically another lattice invariant rearrangement of the atoms minimizing the strain could be realized via slip, however, in SMAs twinning is the preferred mechanism. This circumstance has been expressed in the introductory classification of the crystallographic theory by stating that the slip systems have to be sufficiently rigid. In the literature also other examples referred to as self-accommodated beside twinned martensite are found, where the internal structure of the shape-memory related martensite is not a twinned arrangement, but consists of long period stacking order modulations [23, 24]. It is pointed out that in this work the theory behind the deformation of the martensite domain is only related to twinned martensite. A general difference between slip and twinning should be mentioned at this point: While atoms are moving at least one atomic distance performing a slip operation this is not the case for twinning, where atoms move only a fraction of an atomic distance. In a mere temperature dependent transformation the arrangement of variants occurs such that the average macroscopic shape change is negligible, because the lattice is differently orientated within each grain and the tension generated from one variant is compensated by another minimizing the total energy.

If a partially or completely transformed alloy is deformed, the strain minimizing, accommodated martensite will be stressed further. In principle the variants rearrange again,

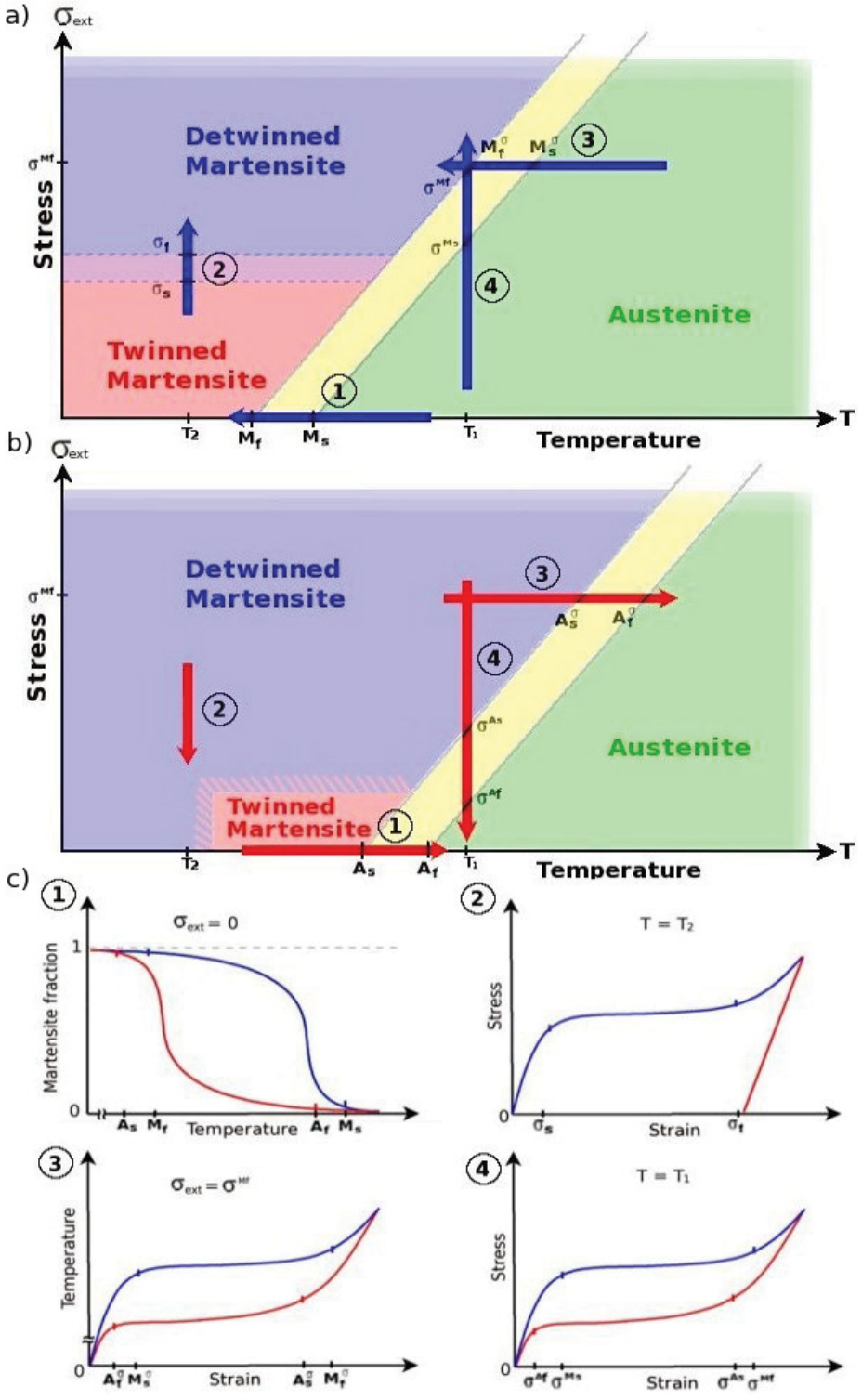

Figure 1.4.: a) and b): Schematic possible transformation paths in a shape-memory alloy. Forward and reverse transformation temperatures are shifted and also depend on the applied external stress. Hysteretic behavior of each individual transformation path is given in c), where the color of the arrow in the above phase diagram matches the line color in the hysteresis curves.

if possible, so as to remain stress-free. From an energetic point of view: the stress state energetically favors the rearrangement of the variants. The mechanism is the following: Microscopically, a resolved shear stress according to Schmid's law [25] acts on the twin plane. When the resolved shear stress reaches a critical value, the variant preferred by orientation and direction of stress will evolve at the expense of other variants. At the grain level this process is called detwinning, whereas on a macroscopic scale the material's behavior is termed pseudoelastic or superelastic, describing its elastic and reversible response to an applied stress. The mechanical load results in a macroscopic strain in the direction of the load that is limited due to the compatibility of variants and orientations and is therefore found to be higher in materials with more possible variants or a favorable texture. The resulting deformation appears macroscopically plastic, because there is not enough restoring force once a new energy minimizing state has been reached since the variants in their new configurations are not much more stressed than before (Figure 1.4,c.2). Locally re-transforming would only lead to an increase of strain energy. The mechanical load on twinned martensite can lead to detwinning processes resulting in a macroscopic shape change that is retained if the load is released. However subsequently heating the material above its transformation temperature lets each variant of martensite transform back into austenite, completely recovering its original state at A_f, as is shown in Figure 1.5. Also, additional cooling below M_f again leads to twinned martensite with no preferred variant. To trigger detwinning the load must be sufficiently large. However the stress level for reorientation of the variants is still far lower than the plastic yield stress of martensite. The transformation temperatures are dependent on the magnitude of the applied load (regardless of tension or compression) because it is the second largest contribution besides the temperature dependent chemical potential in the free-energy stabilizing the martensite at higher temperatures, as is explained in section 2.6. A higher magnitude of applied load leads to higher transformation temperatures. The new transformation temperatures are represented as M_σ^f, M_σ^s, A_σ^s, A_σ^f, where σ refers to the magnitude of a uniaxial stress state or an appropriate scalar measure for a multiaxial stress state [26]. Following the above explanation the temperature interval for pseudoelasticity depends on the magnitude of the applied stress as well. Typically the effect is observed about 0-40 K above the A_f temperature. In this temperature interval removing the mechanical load causes NiTi to spontaneously spring back to its original shape. In this mode the NiTi possesses an elastic range 10-30 times greater than that of a normal spring material. The minimum amount of stress which is required for the detwinning procedure to start is called the detwinning start stress (σ_S), and the maximum level of stress that results in a complete detwinning of the material is called the detwinning finish stress (σ_F).

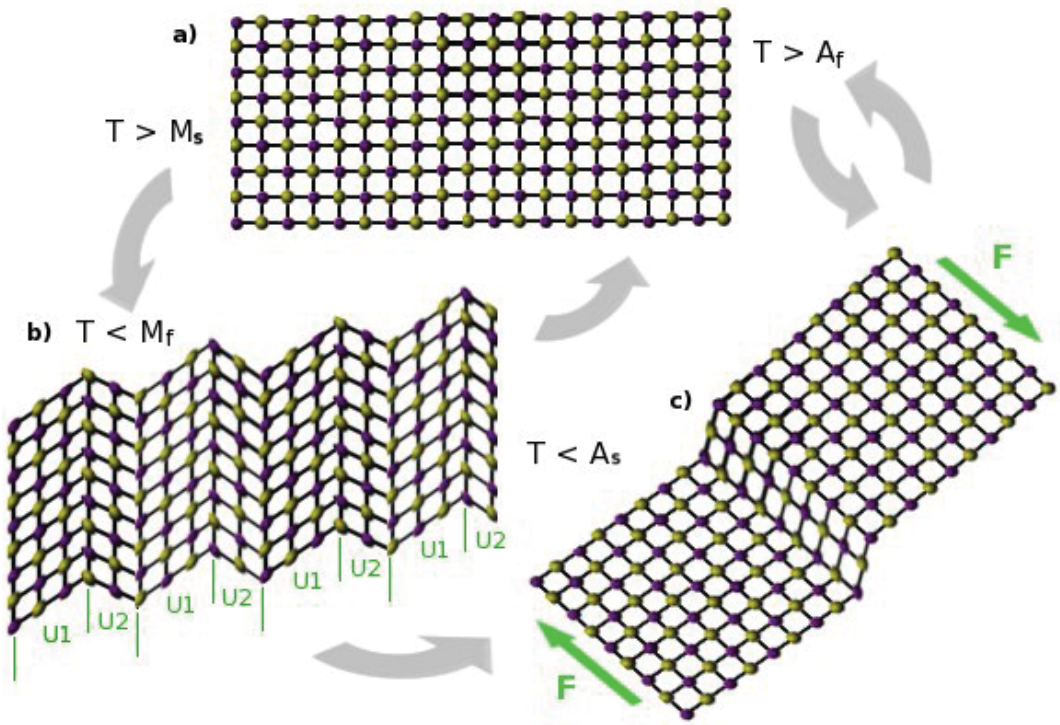

Figure 1.5.: Shape-memory effect: Upon cooling austenite a) starts transforming at temperatures $< M_s$ to twinned martensite. Fig. b) shows fully twinned martensite consisting of two variants U1 and U2. Applying a mechanical load causes one variant to grow at the expense of the other c). Heating above A_f recovers the original, undeformed austenite state.

Summarizing, martensite can form in two ways: twinned martensite, which is formed by a selection of self-accommodated twinned variants, and detwinned or reoriented martensite in which a specific variant is dominant. In which way it will form depends on the stress state during or after the transformation. The pseudoelastic behavior of the stress-induced transformation leads to strain generation during loading and subsequent strain recovery upon loading at temperatures above A_f. This process is energy dissipating and therefore accompanied by hysteresis, which can be explained in the following way: The reverse martensite-austenite transformation is biased by the elastic energy stored during the forward austenite-martensite transformation. In other words, the energy assists the chemical driving-force gained by heating. A_s a consequence, the As temperature for the reverse transformation frequently lies below the M_s temperature marking the start of transformation on cooling. The hysteresis resulting during a pseudoelastic loading and unloading cycle is a measure of the damping capacity of a vibrating device fabricated from a shape-memory material cycled under extreme stress conditions exceeding the critical stress needed to induce martensite. Interestingly, vibrating fully martensitic samples also exhibit high damping due to friction of the forward-backward moving domain boundaries. Generally, repeated thermomechanical cycling, that is deformation in martensitic condition followed by heating and cooling called “training”, causes permanent changes in the material behavior due to the formation of a characteristic dislocation structure which

is responsible for the TWSME. Pure thermal cycling causes a draft of the transformation temperatures, whereas mechanical cycling on the one hand decreases the necessary transformation stresses and on the other hands leads to higher irreversibility.

If one wants the material to recover all of the macroscopic strain upon heating this is only possible if all strains were produced by deformations that are mechanically reversible such as: Elastic deformation, reversible-growth of martensite, twinning, movement of stacking fault partials and slip by superlattice dislocation in crystals with long-range order [23]. All irreversible deformation such as (i) irreversible growth of martensite, high temperature creep, relaxation processes of dislocation configurations and non-planar slip cannot be recovered. The only known mode of reversible deformation accommodating the lattice shape-change after forming martensite in nano crystalline NiTi is elastic strain which is also the basis of the following model. Recoverable strains with at least 5% and a maximum of 16% are reported in the literature in extreme cases depending on the specimen shape, testing conditions, microstructure and other factors. Finally Otsuka [20] pointed out that small diffusive contributions are of great importance for the enhancement of the shape-memory effect.

1.1.3. Consequences of a Three Dimensional, Multi-Grain Model

Each simulation should start at a simple level to better understand the changes occurring after extending an already existing model. In the case of martensitic transformations in NiTi on a nanoscale a single grain model may be sufficient for the prediction of a certain morphology. It lacks however the interactions of grains among each other. Especially the above stated effect of self-accommodation on the one hand and self-triggering of the martensitic variants on the other hand can only be obtained in a multi-grain model. Furthermore, at the nanoscale several effects occur in a relatively small grain-size range (see subsection 1.2.2) which may be explained by the fact that all dimensions are near atomic dimensions. To explain the variety of effects an exact representation of energy quantities at this scale is crucial. In three dimensions surface and strain energies differ more between various transforming regions than in 2D due to the different scaling behavior of volumes and surfaces compared to areas and lines in two dimensions. Also the complexity of the stress state may not be sufficiently reconstructed in two dimensional models. Strain energy contributions, e.g., might be underestimated. In 3D coherency after deformation obviously requires more constraints. For all these reasons for numerically analyzing martensitic transformations a three dimensional multi-grain model is considered necessary.

1.2. NiTi

1.2.1. Metallurgy

NiTi is an ordered intermetallic compound. It was discovered in 1963 in the Naval Ordnance Laboratory (NOL) which led to its commercial name NiTiNOL. Mostly, NiTi is produced in a nearly fifty-fifty at.% composition, since only in the vicinity of this composition the martensitic transformation required for its technological applications is obtained. NiTi's homogeneous high temperature phase with a bcc crystal lattice is called B2 according to its Strukturbericht designation, i.e. a classification that summarizes crystal structures belonging to the same space group (having the same symmetry), where additionally the same points in the unit cell are occupied. If centered atoms and cornered atoms are identical in a bcc lattice, then also the structure is said to be bcc However, in NiTi they are different, therefore NiTi's structure is called B2. The B stands for a compound of the stoichiometric type AB and the number two for the caesium chloride structure. Nowadays this designation is mainly used in metallurgy. A short explanation of crystal structures, point and space groups as well as symmetries is given in section 2.1. The main difficulty of the fabrication of NiTi is the exceptionally tight compositional restriction in which it is stable or metastable at room temperature, as can be seen in its phase diagram in Figure 1.6 according to Masslaski, Otsuka [27, 28] and Bastin [29], respectively. Above all the high reactivity of titanium especially with oxygen and carbon shifting the composition in Ni's favor is a crucial problem to deal with. Deviations from its stoichiometric composition are due to vacancies and substitutions of the two elements. For Ni contents above 50%, substitutions of Ni in the Ti sub-lattice and for Ni contents under 50% vacancies and Ti substitutions in the Ni sub-lattice to equal amounts are typical [30]. NiTi's phase diagram has for a long time been discussed controversially, since transformation temperatures are dependent of pre-treatments especially for Ni contents > 50at.%. The material's history in terms of cycling, deformation and precipitation as well as a possible misinterpretation of the R-phase transformation therefore are possible uncertainties in the course of determining the phase diagram. Since in this work only nearly stoichiometrically equal compositions are treated some properties of NiTi's comparable small phase space are listed subsequently:

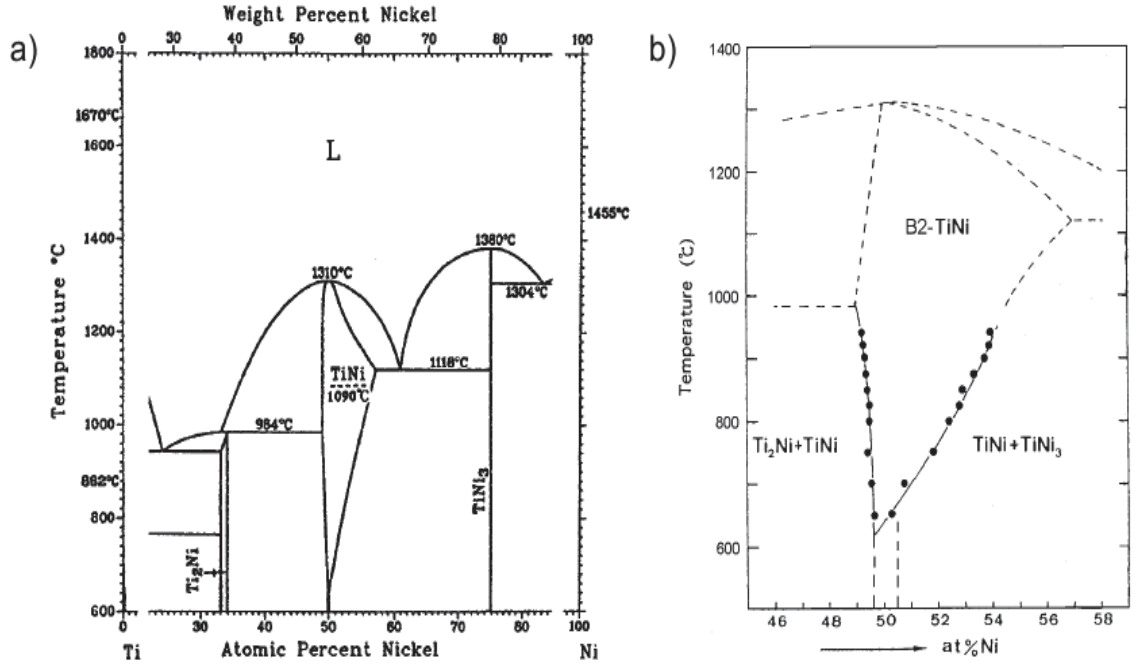

Figure 1.6.: Phase diagram of NiTi: a) due to Masslaski/Otsuka and b) Bastin. Only an almost stoichiometric equally composition is stable at temperatures near room-temperature

- The stoichiometrically equal composition exhibits the maximum A_f temperature of 120°C of all compositions studied [26]. M_s decreases by more than 150K and A_f then lies around -40°C as the Ni content reaches 51%. A variation of composition can change NiTi's room temperature characteristics from a purely thermal SME to pseudoelasticity. A further increase of Ni even causes the martensitic transformation to vanish at about 51.5% [30]. Lowering the Ni content of the balanced composition does not significantly change the transformation temperature.

- The solubility of Ti in the perfect fifty-fifty B2 matrix is less than 0.5 at.%, whereas on the Ni rich side the range of existence of the B2 phase starts to significantly broaden to higher solubility for Ni above 600°C. At room temperature the B2 phase space lies between 50-50,5 at.% Ni. To achieve a then already metastable B2 austenite phase outside a composition range of 50-50.5 at.% Ni at room temperature, the specimen has to be heated until it is fully homogeneous, which is typically at temperatures from 800-1000°C followed by quenching in water.

- For severely plastically deformed specimens of NiTi recrystallization starts around 550°C [31] which must be even lower in the nanograined specimens obtained through high pressure torsion (HPT) regarded in this work, due to the much higher driving-force for recrystallization, i.e. reduction of dislocation energy.

- For compostions above 50at.% Ni aging at elevated temperatures around 400°C leads

to the formation of lenticular Ti_3Ni_4 precipitates. The Ti_3Ni_4 precipitates also act as nucleation sites for martensite, and obstacles for dislocation motion. This mechanism effectively increases the critical stress for dislocation motion and decreases the critical stress for inducing martensite, which has several effects on NiTi's shape-memory effects explained elsewhere [32]. Furthermore the precipitates' stress fields are claimed to enable the formation of an intermediate rhombohedral R Phase. For a long time the R-phase transformation was thought to be a pre-martensitic phenomenon since it appears under certain conditions prior to the martensitic transformation [20]. Normally the inhibition of this phase is under control thus it vanishes in heat treatments at high temperatures and it is generally only associated with certain conditions [33]. Whether the R-phase is desired or not depends on the application of NiTi. On the one hand the R-phase transformation has a very small hysteresis, which sometimes makes it desirable in some actuators, on the other hand it does not offer large shape memory effects.

- Also the most important cases of alloying are mentioned here. Adding Cu to the composition lowers the hysteresis of the SMA response but unfortunately also the transformation strain, and at a Cu content above 7at.% the B2 austenite transforms to the orthorombic B19 structure. The opposite effect, i.e. broadening of the hysteresis can be achieved by alloying with Nb, which is necessary for coupling devices because they are required to show minimal response to wide temperature ranges. So called High Temperature SMAs (HTSMA) are produced by ternary adding Pd, Pt, Hf, Au and Zr. Their transformation temperatures are in the range of 100-800°C [26]. It was also found that adding Co or Fe to the existing NiTi system causes a drastic decrease in transformation temperature, opening new opportunities for its applications as is described in subsection 1.2.3.

Concerning preparation there are two main paths. First there is vacuum electrical arc melting, where no carbon is introduced during melting, which makes it mainly the procedure of choice since the presence of carbon lowers the reversibility of the martensitic transformation and hence the SME. The second way is vacuum induction melting, where alternating magnetic fields are used to heat the raw material in a carbon crucible, making the introduction of carbon inevitable.

1.2.2. Nanostructured NiTi

In a number of experiments enhanced mechanical and functional properties of ultra-fine grained and nano crystalline Nitinol as compared to a coarse-grained material were demonstrated [34, 35]. Therefore the production of a nano structured NiTi SMA is a promising way for the realization of high functional shape recovery properties. Probably the most popular effect accounting for a significant change of mechanical material properties on a nanometer scale is the Hall-Patch effect. This effect basically states that dislocation movement is impeded by grain boundaries. Mathematically this is expressed as $\sigma_y \propto d^{\frac{1}{2}}$, where σ_y is the yield strength and d is the average grain-size. Below a certain grain-size, usually less than 100nm, this is not the case anymore and the yield strength remains constant or even decreases. Explanations for this so called "Inverse Hall-Petch" where given e.g. by Carlton et al [36]. Furthermore, due to the near atomic sizes on a nanometer scale the grain boundary volume increases significantly. Gleiter [37] estimated the specific grain boundary volume for a grain-size of 1 μm to be around 0.2%, whereas at a grain-size of 10 nm this value increases to around 20%. For NiTi an optimum nano grain-size is reported to be located in the range from 40 to 80nm [34], where considerations of a higher true ('dislocation') yield stress against a reduced thermal stability (lower martensite start temperature M_s) are taken into account. Subsequently a short overview on the terminology of grain-sizes frequently found in the literature shall be given. In general, the term ultrafine grain is used for average grain diameters between 1 μm and 2 μm. The term submicron structure (if classified on its own) refers roughly to grain-sizes between 100nm and 1μm and the term nano structure denotes grain-sizes below 100nm. In the case of NiTi a nano grained structure is obtained using high pressure torsion (HPT) resulting in very high hydrostatic pressures [38]. Coarse-grained NiTi is converted to an amorphous phase which transforms at 300-450°C into a nano crystalline structure. Grain-sizes are dependent on the duration of heat treatment and vary in an interval of 5- 350nm [39, 40, 35]. Martensitic transformations of such nano structures show typical size dependent morphologies that are only observed in nanograins. It is reported that under a critical grain-size of approximately 50nm no B19' martensite is found. These grains are preferably composed of retained austenite or R-phase, respectively. Larger grains transforming martensitically by twinning show two different morphologies: It was found that nanograins under a critical grain-size fully transform into a single lamiate of two martensite variants, whereas above that grain-size a so called herringbone structure, i.e. two twinned laminates separated by a junction plane, is the preferred morphology for energetic reasons [41]. As already mentioned in this work only single laminate variants are modeled as the grains are assumed small enough. Figure 1.7 shows transmission electron microscope (TEM) images of a Ni50.3at.%-Ti alloy of a martensite nano grain fully consisting of a single laminate and a herringbone structure respectively. In nanostructured NiTi mainly so called compound

twins are found (for a classification of twins see subsection 2.5.2). In a purely thermal transformation it is observed that twins consist on average of equal amounts (50-50) of two martensite variants ($\mu = 0.5$). Martensitic variants down to a thickness as low as 0.9nm are observed in the TEM approximately corresponding to a range of 6 atoms. As mentioned previously dislocations are only rarely observed, hence plastic deformation plays no role [39].

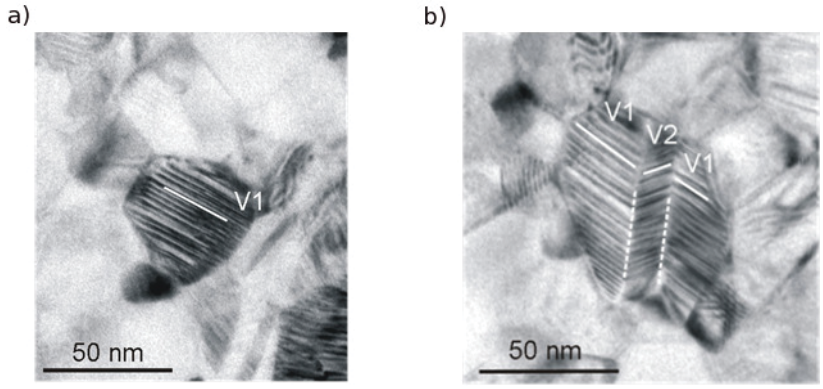

Figure 1.7.: Transmission electron micrographs of martensitic NiTi nanograins. Compound twins of B19' indicated by white lines. a) The whole grain consists of a single laminate. b) Above a critical grain-size a so called "Herringbone" structure consisting of more than one twinned laminate becomes energetically more favorable.

1.2.3. Applications of NiTi

Because of the superior mechanical, chemical and shape-memory properties of NiTi alloys, this alloy system among all SMAs has been applied most successfully. About 90% of present SMA applications use these alloys. Commercial applications of shape memory devices can be divided into four groups [8]:

1. Motion: By free recovery during heating and/or cooling
2. Stress: By constrained recovery during heating and/or cooling
3. Work: By displacing a force, e.g. In actuators and sensors
4. Energy storage: by pseudoelastic loading of the specimen

Subsequently some important practical applications following the above given functionality are given. The majority of applications of NiTi are found in medical engineering. The 50:50 ratio of NiTi has become the alloy system of choice due to bio-compatibility issues. Histoid-spreader, guide wires in catheters and endoscopes, stents, clamps for bones, wires

of braces are only some examples [42]. Once the SMA is at ambient body temperature it contracts back to its original shape, applying a constant force to move the teeth or widen veins. The braces do not need to be retightened as often as conventional stainless steel wires since they can contract when the teeth displace. In aerospace and automotive engineering NiTi is used for light weight design as well as high damping materials to absorb and dissipate energy or damp vibrations in the hysteresis of the reversible transformation without taking damage. Couplings of NiTi are a popular way to replace other kinds of joints for stability reasons, NiTi's high corrosion resistance and in some cases weight considerations. In the galvanic series of metals, NiTi based alloys are slightly nobler than stainless steel and therefore show a comparable corrosion behavior. The first industrial application of NiTi was a coupling device, formerly known as Cryofit in aircrafts in the mid 1960's, where it replaced the weldseam at the area closest to fuel storage units. Over the last decade the demand for actuation under high temperature operating conditions, driven by the aerospace and oil industry, has revived a great deal of interest in the developement of HTSMAs. The oil industry is interested in the actuation capabilities for release devices and protection systems for downhole drilling equipment. Regarding upcoming topic of energy efficiency, NiTi's functional properties can contribute positively to the development of new technologies. They may be able to substitute components like servo-electric actuators, operated at high cycle numbers. The so obtained reduction of weight probably also makes them economically more attractive in other fields of applications. Other useful areas of applications are: valves, in micro-electro-mechanical systems (MEMS), robotics, sensors and heat engines. Applications where cycle transformation fatigue is an issue, are separated in two categories: (i) Functional fatigue means that the material's functional properties like reversible strains change in the course of cycling, whereas (ii) structural fatigue points at the initiation and accumulation of defects that eventually lead to failure due to cracks. Additional advantages of using NiTi SMAs in the engineering area beside the already given ones are its simple, noiseless training mechanism and its high power to weight ratio.

2. Theoretical description

The following chapter deals with the theoretical background of the physical quantities and their relevant relations in this work. In the first section a short introduction to symmetry in general and particular for crystals is given, which is considered necessary to limit the very broad terminology and reveal ambiguities. Symmetry is an important factor in this work since a three dimensional model is considered and symmetry plays an important role for the equivalence of certain states as will be elaborated. A review of mathematics used here is given. Especially different kinds of product definitions are clarified, and basic matrix algebra used in continuum mechanics necessary for this work is treated. Then basic ideas of the non-linear theory of martensitic transformations such as the Cauchy-Born hypothesis and the Erkicksen-Pitteri neighborhood are introduced and discussed. Next, the martensitic transformation in NiTi is discussed with respect to crystal structures, lattice kinematics, martensite variants and classification of twins. Then the governing equations of the twinning theory predicting coherent (smooth) interfaces are presented and discussed, since in this model a certain twinned morphology is considered. The calculation results of the twinning theory are comparted with the experimentally observed ones, which obviously represent energy minimizing states. Note that this work aims to elaborate and compactly summarize twinning theory from different points of view, hence different theories will be discussed and their intersections as well as their differences will be pointed out. However, all these theories refer to an initial single crystal and it is not clear how the arising structures interact in a polycrystal. Therefore, an energy based model described in chapter 3 is created. Finally, energy contributions due to the phase stability are discussed to define a transformation criterion.

2.1. Point Groups, Space Groups, Symmetry and Crystallography

As stated in 1.1.1 in martensitic transformations normally there are definite orientation relationships between the parent and the product phase. It is common practice to describe them by giving the relations of crystallographic planes or directions of the two phases, determined by X-Ray spectroscopy for instance. However, symmetry relations are considered a clearer approach when all possible relationships are being investigated. This section is intended to elucidate the concept of symmetry as well as to clarify some terminology. To start with some basic definitions are given:

- A mapping of an object onto itself is the basis of the concept of geometric symmetry. In mathematics a mapping is called an isometry if it leaves all distances invariant. An isometry is a special kind of an affine mapping, in which parallel lines are mapped to parallel lines, lengths and angles may be distorted but fractions of lengths on the same line are preserved.

- A symmetry operation is an isometry which maps the object onto itself. This does not mean that each point of the object is mapped onto itself, rather the object is mapped in such a way that an observer cannot distinguish the states of the object before and after the mapping, therefore the object is said to be left invariant.

- A crystal is a finite block of a periodic array of atoms. The smallest part containing all the information on the positions of atoms relative to each other is called a unit cell. The two main rules for the selection of a unit cell with decreasing importance are (i) the highest possible symmetry and (ii) the smallest configuration possible. If ambiguity exists nevertheless, the last criterion for a unit cell is simplicity.

All symmetry operations for a unit cell can be divided into proper rotations around a certain axis and improper rotations or rotoinversions, which are a combination of a rotation with an inversion at a point. All the rotation axes (proper and improper) must pass through the center of the object, hence there is always at least one point that remains invariant, while moving other directions and faces of the crystal to positions and directions of the same kind. The existence of an invariant point is the reason why the set of all symmetry operations of a finite object is called a “point group”. There are only 32 possible crystallographic point groups, or in crystallographic language “crystal classes”. Space groups in general are symmetry groups of a configuration in space (subgroup of the Euclidean motion group). In crystallography these three dimensional configurations are restricted to ideal, periodic crystals and are called crystallographic groups. A crystallo-

graphic group extends the symmetries of point groups by translational symmetries (pure, glide plane, screw axes), where translational symmetries are transforms which map the crystal pattern onto itself upon translation. As the name space group suggests there are no invariant points, since the lattice translations always present do not leave any point unmoved. In crystallography space groups are divided into 7 crystal systems according to their point groups, and into 7 lattice systems according to their translational symmetry. Limiting oneself to unit cells there are exactly 14 unique types called the Bravais lattices. In other words, one can show that there are seven distinct point groups that arise in Bravais lattices [9], and they describe the seven different symmetry types (lattice systems) shown in Figure 2.1. Note that two lattices only belong to the same Bravais type if and only if they coincide both in their point-group and in their centering.

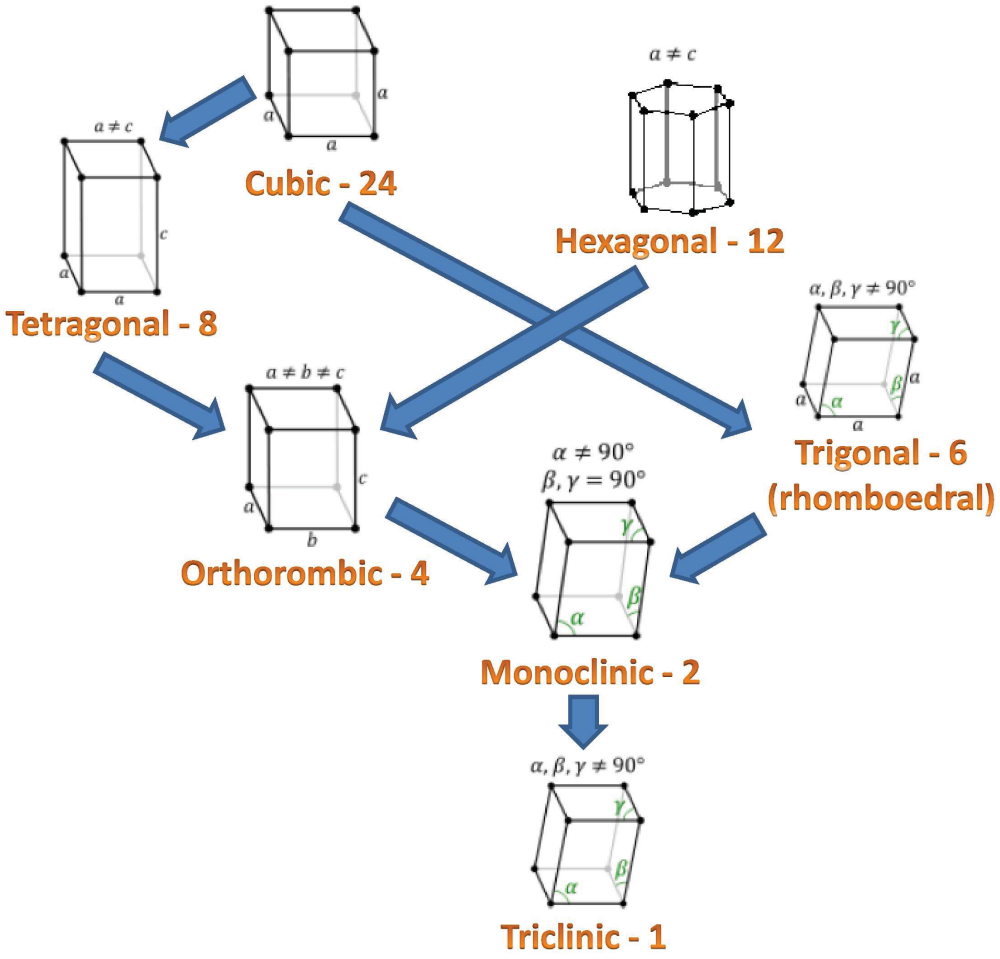

Figure 2.1.: The seven types of lattice systems. The numbers after the system name indicate the number of rotational symmetries mapping the lattice onto itself. The arrows indicate that lower systems are subgroups of the higher ones, where the cubic system has the highest and the triclinic the lowest symmetry. Note that the hexagonal lattice is not a subgroup of the cubic lattice.

A crystal's rotational symmetry is restricted to five types of symmetry axes denoted as 1,2,3,4 or 6 corresponding to their order, i.e. the number of consecutive elementary rotations that will restore the object to its initial orientation. This is a consequence of the periodicity of crystal lattices. Generally a symmetry operation of a crystal pattern is called a crystallographic symmetry operation.
One of the main tasks of theoretical crystallography is to sort the infinite number of conceivable crystal patterns into a finite number of classes, where the members of each class have certain properties in common. In such a classification, each crystal pattern is assigned to only one class. The elements of a class are called equivalent, the classes being equivalence classes in the mathematical sense of the word.

2.2. Mathematics

This section introduces the mathematical framework used throughout this work. Especially, different definitions of products of tensors in the form of dyads and matrices are elaborated and the differences between certain notations are shown. Also some matrix algebra commonly used in continuum mechanics is presented.

2.2.1. Terms and Definitions of Products of 2nd Order Tensors and Matrices

Equations of continuum mechanics describe displacements, strains and stresses by vector and tensor fields. Thereby, four notational systems are in common use. This may be confusing since for each notation the operations that may be generally the same in the case of continuum mechanics are often termed differently. This section is intended to give a quick overview on the terms used for these notations especially for the different kinds of mathematical "products". (i) The first notation is index or component notation, where an index identifies components of vectors and tensors. It has convenient abbreviation rules, such as Einstein's summation convention [43], and can handle arbitrary tensors of any order and coordinate system as well as nonlinear expressions. When used in non-Cartesian coordinates, it sharply distinguishes between covariant and contravariant quantities, which basically describe the behavior of a tensor upon a transformation of coordinates [43]. However, in the case of Euclidean distance metrics, where the dual space and the original vector space are identical, co- and contravariant quantities are also identical, hence the stress tensor can also be written in matrix form. (ii) The second one is direct notation, where vectors and tensors are represented by single symbols, usually bold characters, which are linked by well known operators of mathematical physics, such

as " $\cdot$ " for a dot product or " $\times$ " for the vector cross product. On the one hand it is beneficial for a quick visualization while on the other hand some operations become undefined. (iii) Third is matrix notation, which is similar to the direct notation with the difference that mathematical expressions are appropriately rewritten so that only matrix operations can be used. A disadvantage is the loss of physical meaning. For example the symmetric second-order stress tensor is recast as a 6 entry vector for finite element method (FEM) applications. In vector form one cannot simply define eigenvalues such as the principal stresses of the stress tensor. Finally, there is of course a full notation, where every term is spelled out and no ambiguities of interpretation can arise, but in which for instance equation (3.6) would not fit on one page.
Starting from two vectors $\vec{a} = a_1\vec{x}+a_2\vec{y}+a_3\vec{z}$ and $\vec{b} = b_1\vec{x}+b_2\vec{y}+b_3\vec{z}$ in 3D cartesian space, subsequently the dot product = scalar product (= inner product in a general perspective as tensors in Euclidean space) of these two vectors is written in all four notations:

$$\underbrace{a_i b_i}_{index} = \underbrace{\mathbf{a}\cdot\mathbf{b}}_{direct} = \underbrace{\mathbf{a}^\mathbf{T}\mathbf{b}}_{matrix} = \underbrace{a_1b_1 + a_2b_2 + a_3b_3}_{full} \tag{2.1}$$

A vector can be regarded as a tensor of rank 1 and a scalar as a tensor of rank 0. From this perspective the dot product is an algebraic operation that reduces the rank of two first order tensors. In general the single dot product between a tensor of order n and a tensor of order m is a tensor of order n + m - 2. This concept can be extended until the rank of the product is 0, whereby a commonly used notation is to write as many dots horizontally next to each other as the rank of the tensor. This is also called index contraction. Also index based notation using Einstein's sum convention is used causing equal indices to disappear. Both notations are subsequently written for second order tensors.

$$\underline{\underline{A}} : \underline{\underline{B}} = A_{ij}B_{ij} \tag{2.2}$$

In the case of matrices the equivalent operation is called Frobenius inner product and is written analogously to the (vertical) double dot product.
Next the two starting vectors $\vec{a}$ and $\vec{b}$ are multiplied in distributive fashion. The result is called a dyad and the product dyadic product. A dyad can be represented as a square matrix using the coefficients of the unit dyads. A dyadic product sometimes is written as the juxtaposition **ab** as established by Gibbs in 1884. The dyadic product is not commutative in general. It can be considered a special case of the general tensor product denoted by "$\otimes$". Subsequently several notations are given:

$$\mathbf{ab} = \mathbf{a}\otimes\mathbf{b} = \mathbf{ab}^\mathbf{T} = a_i b_j = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix} = \begin{bmatrix} a_1b_1 & a_1b_2 & a_1b_3 \\ a_2b_1 & a_2b_2 & a_2b_3 \\ a_3b_1 & a_3b_2 & a_3b_3 \end{bmatrix} \tag{2.3}$$

The dot product of a dyad and a vector will be especially important.

$$(\mathbf{a} \otimes \mathbf{b}) \cdot \mathbf{c} = \mathbf{a} \otimes (\mathbf{b} \cdot \mathbf{c}) = (\mathbf{ab}) \cdot \mathbf{c} = \mathbf{a}(\mathbf{b} \cdot \mathbf{c}) \tag{2.4}$$

As can be seen any vector $\mathbf{c}$ is converted to a vector which is parallel to $\mathbf{a}$ and whose magnitude multiplied by the scalar $\mathbf{b} \cdot \mathbf{c}$. It should be noted, that this is not commutative. By exchanging the vector and the dyad, the result is a vector which is parallel to $\mathbf{b}$ and whose magnitude is multiplied by the scalar $\mathbf{a} \cdot \mathbf{c}$. The dyadic product will play a crucial role in the discussion concerning coherency. The linear combination of dyads is called a dyadic. The same way a dyad is formed from two vectors, a triad is formed from a dyad and a vector and a tetrad is formed from two dyads, whereby every formed tensor of rank n has 3^n components in three-dimensional space. As mentioned a dyad can be written in matrix form. In matrix algebra the dyadic product of two dyads is known as the Kronecker product. It converts two matrices of arbitrary size to a block matrix. For example if $\mathbf{A}$ and $\mathbf{B}$ are 3×3 matrices it gives:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & a_{13}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & a_{23}\mathbf{B} \\ a_{31}\mathbf{B} & a_{32}\mathbf{B} & a_{33}\mathbf{B} \end{bmatrix} \tag{2.5}$$

In full notation this would give a 9×9 matrix. It should be noted that there is no index notation for the Kronecker product. However, since vectors can be seen as matrices with the dimensons $1 \times$ n the Kronecker product can be coded as an outer product of those vectors. Finally, the well known matrix product is discussed. The matrix product is very important for linear transformations as in Equation 2.13 and linear equations. In general it is not commutative. The general definition of the matrix product is:

$$(\mathbf{AB})_{ij} = A_{ik}B_{kj} \tag{2.6}$$

Thus the product $\mathbf{AB}$ is defined only if the number of columns in A is equal to the number of rows in B, in this case N. This also works for vectors written as matrices. Given two column vectors $\mathbf{a}$ and $\mathbf{b}$, the Euclidean inner product and outer product are the simplest special cases of the matrix product by transposing one column vector into a row vector respectively as can be seen in (2.1) and (2.3).

2.2.2. Fundamental Algebra in Continuum Mechanics

In this section the notion of the deformation gradient and some important algebraic relations commonly used in continuum mechanics are given. Subsequently, some relations necessary for the definition of the so called polar decomposition theorem are presented.

- A rotation or reflection is defined by all matrices for which the following holds: $\mathbf{Q^T Q = Q Q^T = I}$ and det$\mathbf{Q}$ = 1 is a rotation or det$\mathbf{Q}$ = -1 is a reflection.

- If a matrix $\mathbf{U}$ is symmetric ($\mathbf{U = U^T}$) then it has 3 real eigenvalues and one can choose corresponding eigenvectors to be mutually perpendicular unit vectors.

- A matrix $\mathbf{U}$ is positive definite if $a_i U_{ij} a_j > 0$ for all vectors $\mathbf{a} \neq 0$. A positive definite matrix has only positive eigenvalues

For any given matrix $\mathbf{F}$ with det($\mathbf{F}$) > 0 there exists a unique rotation $\mathbf{Q}$ and a unique positive-definite symmetric matrix $\mathbf{U}$ (right stretch tensor) such that $\mathbf{F = QU}$, where $\mathbf{Q}$ and $\mathbf{U}$ can be calculated by:

$$\mathbf{U} = \sqrt{\mathbf{F^T F}} \tag{2.7}$$

$$\mathbf{Q} = \mathbf{F U^{-1}} \tag{2.8}$$

In the context of quantities describing martensitic transformations $\mathbf{U}$ is a pure stretch and is called Bain strain. This decomposition will be crucial for the discussion of martensitic variants in 2.4.3 and that of frame indifference in 2.3.

Now the general definition of the deformation gradient is introduced. Given any deformation $\mathbf{y}$ that acts on a vector $\vec{x}$ representing an arbitrary point in three dimensional space, the deformation gradient normally denoted as $\mathbf{F}$ is the matrix of partial derivatives:

$$\mathbf{F} = [\nabla \mathbf{y}(\vec{x})]_{ij} = \frac{\partial y_i}{\partial x_j} \tag{2.9}$$

For a homogeneous deformation the deformation gradient is constant in space, i.e. independent from the spatial coordinates, whereas for an inhomogeneous it is not. The definition above implies that the deformation gradient can be written in a matrix form as for instance the deformation that describes the martensitic transformation from B2 to B19' given in Equation 2.13. A useful feature of the transformation gradient is that the determinant of the deformation gradient describes the local change in volume $det(\mathbf{F}) = \frac{dv}{dV} \overset{F=const}{=} \Delta V$. Note that $det(\mathbf{U}) = 1$ is a necessary and sufficient condition for self-accommodation in SMAs with a cubic austenite [21].

Probably the most important relation in crystallographic theories of martensitic transformations is that of kinematic compatibility upon deformation, which essentially explains the resulting martensitic morphology. In the process of twinning the lattice gets sheared in opposite directions on each side of an invariant plane. Pure shear is a homogeneous deformation, hence the entire transformation around is homogeneous too. However, the deformation gradient is not homogeneous in the area around the habit plane. More pre-

cisely, it jumps across the coherent habit plane. It can be shown that if the deformation is continuous the jump of the deformation gradient must satisfy certain conditions [18]. It is then possible to show that the invariant plane and the deformations on each side can be described by the following relation.

$$\mathbf{F} - \mathbf{G} = \mathbf{a} \otimes \hat{\mathbf{n}} \tag{2.10}$$

where $\mathbf{F}$ and $\mathbf{G}$ are the matrices that describe the homogeneous deformation (shear in the case of martensitic transformations) on each side of the habit plane, $\hat{\mathbf{n}}$ is the normal unit vector to the habit plane and $\mathbf{a}$ is proportional to the amount of shear (amplitude of the jump). This equation is known by many names such as invariant plane condition, kinematic compatibility condition, rank one condition or Hadamard jump condition.

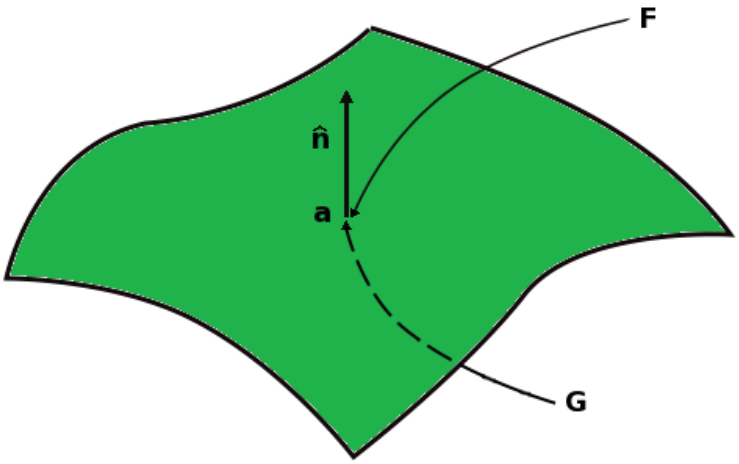

Figure 2.2.: Viszualization of the components in the Hadamard jump condition. $\mathbf{F}$ and $\mathbf{G}$ are the homogeneous deformations on each side of the undistorted interface.

Next, the rotation of a matrix from one orthogonal coordinate system to another with the same origin is discussed. For a given matrix $\mathbf{A}$ in the coordinate system $\mathbf{x}_i$, $\mathbf{A}'$ in another coordinate system $\mathbf{X}_i$ can be obtained by

$$\mathbf{A}' = \mathbf{R}\mathbf{A}\mathbf{R}^{-\mathbf{1}} \overset{\text{orthogonal}}{=} \mathbf{R}\mathbf{A}\mathbf{R}^{\mathbf{T}} \tag{2.11}$$

where $\mathbf{R}$ is the transformation matrix between the initial and the new coordinate system containing the directional cosines of the two orthogonal basis sets with the same origin. Note that the reverse transformation is $\mathbf{A} = \mathbf{R}^{\mathbf{T}}\mathbf{A}'\mathbf{R}$, which is directly obtained from (2.11) by premultiplying both sides by $\mathbf{R}^{\mathbf{T}}$ and postmultiplying both sides by $\mathbf{R}$ giving $\mathbf{R}^{\mathbf{T}}\mathbf{R}\mathbf{A}\mathbf{R}^{\mathbf{T}}\mathbf{R}$ and using $\mathbf{R}^{\mathbf{T}}\mathbf{R} = \mathbf{I}$.

2.3. Continuum Theory of Crystalline Solids

In this section the link between the crystal lattice and the macroscopic continuum is presented following Bhattacharya [21]. In a solid state transformation the Bravais lattice defined by the set of lattice vectors $\mathbf{x_i}$ changes to another Bravais lattice with the basis $\mathbf{X_i}$, where the deformation is described by a deformation gradient $d\mathbf{x} = \mathbf{F}d\mathbf{X}$. As a consequence of symmetry, some deformations map the lattice back onto itself. In continuum theory deformations are normally limited to those that preserve orientation ($det(\mathbf{F}) > 0$). Therefore, as opposed to point groups, reflections are excluded since they do not preserve the orientation. Furthermore, since in shape memory alloys plasticity is very limited, lattice invariant shear deformations associated with plasticity and slip are excluded as well. Eventually, the remaining set of deformations mapping the Bravais lattice back onto itself is the set of rotations describing the point group of the initial Bravais lattice of austenite $\mathcal{P}^a(\mathbf{x_i})$. In the case of NiTi the intial Bravais lattice is of cubic type and the set of rotations mapping the cubic lattice onto itself contains 24 rotations. The next step linking the lattice to the continuum is the Cauchy-Born hypothesis, which states that the continuum also deforms according to the deformation gradient $d\mathbf{x} = \mathbf{F}d\mathbf{X}$. The Cauchy Born hypothesis is a definite way to bridge the atomistic scale and the macroscopic continuum picture. It should however be noted that it only holds as long as the reference configuration and the deformation gradient are homogeneous (i.e. plasticity and slip would violate the Cauchy Born hypothesis). For SMAs it is important to note that the hypothesis is justified. A comprehensive review on the hypothesis and its validity can be found in [17]. Subsequently, some considerations on the energy density are made. Starting from the intital Bravais lattice the stored energy density g is assumed to depend on the lattice vectors and the temperature $g(\mathbf{x_i}, T)$. Furthermore g is assumed to satisfy two properties: (i) Frame-indifference: If the lattice is rotated or a change of frame is carried out in its symmetry group, g does not change. (ii) Material-symmetry: For the same Bravais lattice g is independent from the set of lattice vectors.

$$g(\mathbf{FR}, T) = g(\mathbf{F}, T) = g(\mathbf{RFR^T}, T) \quad \text{for all rotations } \mathcal{P}(\mathbf{x_i}) \tag{2.12}$$

Recall that the goal is to construct a theory where lattice deformations are large enough to include transformations and elastic deformations, but small enough to exclude plasticity and slip. Therefore the idea of a so called Ericksen-Pitteri Neighborhood is introduced, primarily based on three observations: (i) Starting from a specific Bravais lattice the symmetry is easily reduced with small distortions and normally it takes a higher amount of distortion to increase the symmetry. (ii) For small distortions the elements of $\mathcal{P}(\mathbf{x_i})$ map lattices that are close to the parent lattice $\mathbf{x_i}$ to other lattices that are close to $\mathbf{x_i}$, i.e. lattices which can be obtained from $\mathbf{x_i}$ with another small distortion. (iii) Equally

to (ii) $\mathcal{P}(\mathbf{x_i})$ maps lattices that are far away from $\mathbf{x_i}$ to other lattices that are far away. The Erickson-Pitteri neighborhood thus is the set of all lattice vectors that are obtained by a small deformation of the inital set $\mathbf{x_i}$. It entails that only transformations where the symmetry of the parent lattice is a higher than that of the new one can be studied and excludes large distortions such as lattice invariant shears. These restrictions still allow to study martensitic phase transformations in SMAs.

Mathematically the difference of two sets of lattice vectors except for a rotation is conveniently described the metric tensor: $C_{ij} = \mathbf{x_i} \cdot \mathbf{X_j}$. Generally, the same lattice can be described by a variety of basis sets. However, provided that orientation is preserved the initial set $\mathbf{x_i}$ and the new set $\mathbf{X_i}$ describe the same lattice if $\mathbf{x_i} = \mu_i^j \mathbf{X_j}$ where μ_i^j is a 3x3 matrix with a determinant equal to 1. $\underline{\underline{C}}^{\mu} = \underline{\underline{C}}^0$ only if μ_i^j generates an element of the point group. Due to (ii) such μ_i^j also map a metric $\underline{\underline{C}}$, which is close to $\underline{\underline{C}}^0$ to a metric $\underline{\underline{D}}$ close to $\underline{\underline{C}}^0$ and according to (iii) μ_i^j that are not in the point group map a metric $\underline{\underline{C}}$ that is close to $\underline{\underline{C}}^0$ to one that is far away from $\underline{\underline{C}}^0$ (see Figure 2.3 a)). Additionally, since (2.12) also the free energy depends on this metric as shown in Figure 2.3 b). At this point it shall be mentioned that the reason for considering closed groups is that the constitutive function for the free energy of the crystal is assumed to be smooth. Then, following Ball and James the wells can be visualized as circles in space of the deformation gradients as is shown in Figure 2.3 c). The idea is that the free energy density has a multiwell character

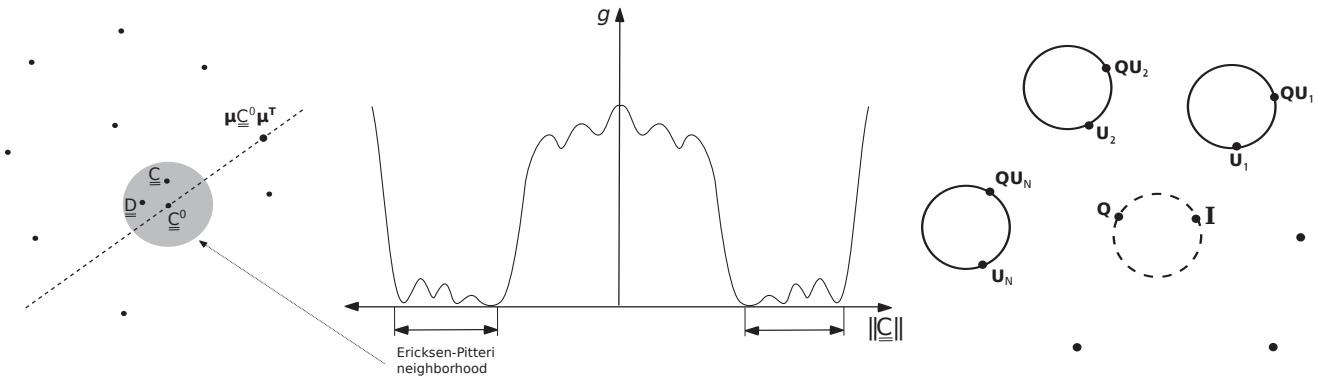

Figure 2.3.: a) Schematic illustration of the Erickson-Pitteri neighborhood in the space of metrics between two lattice vector sets. b) Schematic free energy density function for all metrics along the dashed line in a). c) Representation of energy wells: The circles schematically represent the pre-multiplication with all rotations. The dashed circle is the austenite well and the others are martensite wells .

related to the crystal structures of the involved phases in the transformation. At the transformation temperature, the energy has wells in the space of deformation gradients corresponding to both the austenite and the martensite phases. The position of the wells is determined by the lattice parameters of the crystal lattices of both phases. It shall be mentioned that multi-lattices (combined Bravais lattices) can be treated as the union of multiple Bravais lattices [21].

2.4. Martensitic Transformation in NiTi

In ordered alloys such as NiTi, it has been shown that the phase formed by the martensitic transformation is also ordered. This section presents the atomic structures of NiTi relevant in this work. Its lattice parameters are discussed and the Cartesian coordinate system used throughout this work is presented. In this system first the deformation gradient and subsequently symmetry related so called martensitic variants are represented.

2.4.1. Crystal-Structure of NiTi

As already stated in subsection 1.2.1 the austenitic phase is of an ordered bcc B2 type. Note that the space group of a pure metal with bcc structure is different from B2. The cubic unit cell has a lattice constant of $a_c = 3.015$Å. This lattice constant was determined in a Ni49.75a%-Ti compositon by Otsuka [20]. Figure 2.4 a) shows this unit cell as well as the orientation relation between the cubic lattice of the austenite and a tetragonal unit cell that upon forward transformation will become the monoclinic unit cell of B19' martensite. The monoclinic B19' martensite unit cell has lattice parameters: $a_m = 4.646$Å, $b_m = 2.898$Å, $c_m = 4.108$Åand the monoclinic angle $\gamma_m = 97.78°$. The lattice parameters for the B19' structure were determined in a Ni49.2a%-Ti by Kudoh et al. [44]. Note that the lattice parameters are defined in correspondence to the elastic tensor calculated in a different coordinate system, see [45]. This is important in order to ensure that the calculated deformations and the elastic tensor refer to the same coordinate system. A selection of possible Cartesian coordinate systems for the monoclinic system is discussed separately in the next paragraph. Obviously in an ordered crystal like NiTi the deviation of lattice constants due to varying amounts of Ni and Ti as well as additional alloying elements can abruptly change the material behavior. The here encountered difference of 0.5a% is however considered a reasonable approximation. Finally, some information is given on the so-called R-phase with Strukturbericht-designation B19, which is often confused with B19'. This orthorombic structure is sometimes called 2H according to its stacking sequence ignoring symmetry. The lattice parameters are $b < c < a$ in this space group. This crystal structure has similarity only with the martensite phase in AuCd alloys.

Generally, if a crystal has a single axis of two-fold symmetry, or a single plane of reflection symmetry or both, it belongs to the monoclinic system. The two-fold axis or the normal to the plane of symmetry is called unique axis. In a Cartesian coordinate system, the unique axis is usually assigned to the monoclinic side b and defined as Y axis, and $\beta < 90°$is the angle between +a and +c. Then there are two choices for the X and Z axis namely X||a

or Z||c. For monoclinic systems with m point-group symmetry, there is an alternative choice for the axes. The unique axis is assigned to the side c, which lies along Z, and γ is the angle between the sides $+a$ and $+b$. Also for this case there are two choices for the other two axes, namely X along a or Y along b. As can be seen in Figure 2.4 b) the latter is used in this work. This leads to the following naming convenction compared to Kudoh: $[45] \rightarrow [44] : a \rightarrow c, c \rightarrow b, b \rightarrow a, \gamma \rightarrow \beta$

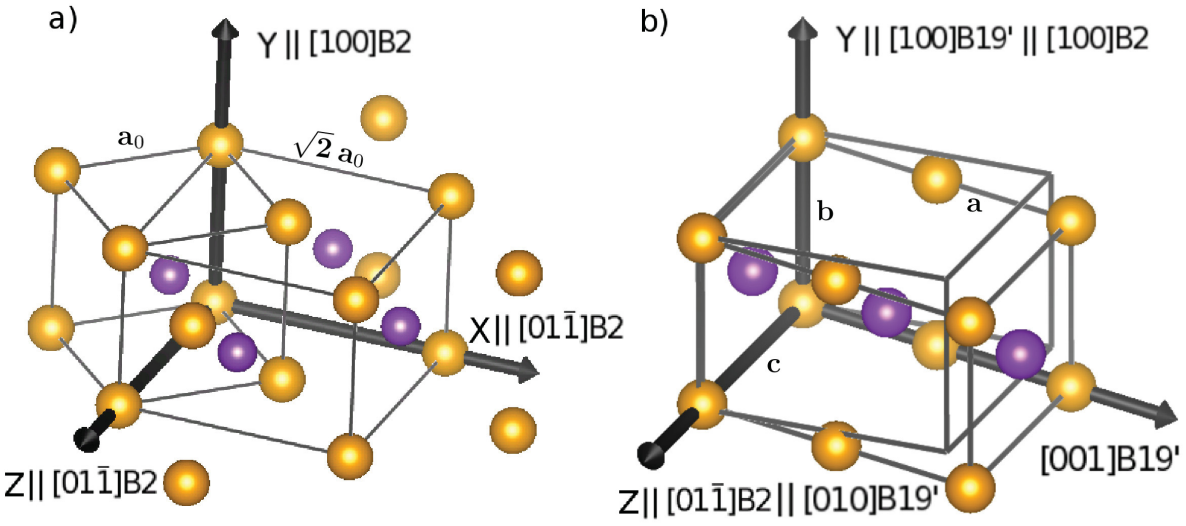

Figure 2.4.: a) Orientation relationship of the parent cubic B2 lattice and a tetragonal cell that can martensitically transform to a monoclinic cell. b) Lattice correspondence of the same tetragonal cell as in a) and a certain monoclinic cell. An additional shuffling of atoms occurs in the middle plane with normal [001]B19'. The monoclinic angle γ is between Y and [001]B19'

2.4.2. Kinematics of Transformation

Pure NiTi specimens directly transform after solution annealing during cooling to martensite. There are plenty of alloys that have a B2 parent structure and transform upon cooling to another structure, however the B2 to B19' transformation is unique. While other alloys change their phase only by changing the stacking-fault order of their (110) planes, in NiTi an additional shear perpendicular to these planes takes place. The unit cell of B19' is stretched against the austenitic matrix by around 10%, whereas the volume barely increases upon transformation at about 0.5% [46]. Macroscopically the deformation can be decomposed into a stretch normal to the habit plane and a shear component almost parallel to it. The latter is called macroscopic shear and quantifies the shape deformation, whereas the former represents the volume change. The habit plane is not distorted, hence the macroscopic shape change associated with the transformation is also called an invarinat plane strain deformation (IPS). Also many other martensitic transformations

show an IPS deformation consisting of an extension and a simple shear. The habit plane and the direction of macroscopic shear are, with few exceptions, not simple low-indexed crystallographic planes or directions of the parent or product phase. Mathematically the deformation of the unit cell can be represented as a homogeneous linear transformation, which is also termed affine transformation. Straight lines remain straight lines and planes remain planes upon transformation. An affine transformation can conveniently be written in a matrix formulation given in 2.13. From the tetragonal unit cell shown in Figure 2.4 a) the whole transformation can be decomposed into three steps. (i) An expansion or contraction along the directions X,Y,Z. (ii) A shuffling of the atoms in the middle plane, i.e. in the case of the variant shown in Figure 2.4 b) the plane with normal [001]B19' in the direction Z. This step is required in order to obtain a close-packed structure. However, it is ignored for the deformation gradient hence it does not contribute to the overall transformation strain. Third, a shear in the plane with normal Z. For the here presented case this writes in matrix form as:

$$\begin{aligned}\underline{\underline{F}}^m &= \begin{pmatrix} 1 & 0 & 0 \\ cot(\gamma_m) & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} sin(\gamma_m)\frac{a_m}{\sqrt{2}a_c} & 0 & 0 \\ 0 & \frac{b_m}{a_c} & 0 \\ 0 & 0 & \frac{c_m}{\sqrt{2}a_c} \end{pmatrix} \\ &= \begin{pmatrix} sin(\gamma_m)\frac{a_m}{\sqrt{2}a_c} & 0 & 0 \\ cos(\gamma_m)\frac{a_m}{\sqrt{2}a_c} & \frac{b_m}{a_c} & 0 \\ 0 & 0 & \frac{c_m}{\sqrt{2}a_c} \end{pmatrix} = \begin{pmatrix} 1.0796 & 0 & 0 \\ -0.1475 & 0.9612 & 0 \\ 0 & 0 & 0.9634 \end{pmatrix} \end{aligned} \quad (2.13)$$

where the first term represents the shear and the second one the stretch. The change of volume during the transformation is $det(\mathbf{F}) = 0.9995$ and can therefore be neglected.

Finally a crucial point of the transformation considering the R-phase shall be mentioned. Actually, in NiTi nanograins the transformation from B2 to B19' does not proceed directly, but via the rhombohedral R-interphase so that the transformation sequence is $\mathbf{B2} \longrightarrow \mathbf{B19} \longrightarrow \mathbf{B19'}$. However, the maximal distortion upon $\mathbf{B2} \longrightarrow \mathbf{B19}$ transformation is only around 1 %, whereas for the $\mathbf{B2} \longrightarrow \mathbf{B19'}$ transformation it is 10%. This gives a ratio of strain energies of 1:100, hence the intermediate R-phase can be neglected in models that are dominated by strain energy, such as the one presented here.

2.4.3. Variants of Martensite

Because of the periodicity of the lattice there are more possibilities to draw a tetragonal unit cell like the one in Figure 2.4 which overlaps with a certain cubic unit cell. Additionally considering the shear responsible for the monoclinic angle in both directions leads to 12 martensite (correspondence) variants. Material scientists commonly de-

scribe such orientation relationships in families of directions <> and planes {} in Miller indices. Popular examples of orientation relationships are e.g. the Kurdjumov Sachs (K-S) relation between bcc austenite and tetragonal martensite in low carbon steels (C $< 0.5m\%$: $111_A||110_M, < 110 >_A ||(111)_M$) or the Burgers relation in intermetallic TiAl between the "high-temperature" bcc $\beta-$ and the "low-temperature" hexagonal close packed (hcp) $\alpha-$phase: $(0001)_\alpha||\{101\}_\beta$. In NiTi, the axis of monoclinic rotational (two-fold) symmetry in the martensite is the cubic $\langle 110 \rangle$ direction (edge marked c in Figure 2.4 b)). This transformation is termed cubic to monoclinic I [21] and the variants are termed face diagonal variants because their two-fold axis is along a face-diagonal of the original cubic unit cell. There exists another type of cubic to monoclinic transformation (II) where the axis of monoclinic symmetry is a cubic $\langle 100 \rangle$ (cube-edge variants) direction such as in CuZnAl, not elaborated here any further. For NiTi the full set of orientation relationships of the B19' lattice vectors with respect to the parent cubic structure can be found elsewhere [39]. A more lucid way to calculate all correspondence variants is by the rotational symmetry group of the cubic parent phase, as is subsequently elaborated. If the deformation gradient of the lattice transformation is symmetric than it can be considered as Bain strain. However, if it is not symmetric as the deformation gradient in (2.13) the polar decomposition theorem (2.7) is used to decompose it into a rotation $\mathbf{R}$ and a positive definite symmetric matrix $\mathbf{U}$, which then is considered the Bain strain. This is important since in the theory for the construction of microstructures presented in section 2.5, only the symmetric part of the deformation gradient needs to be considered. Afterwards, the full set of variants can be calculated by rotating the austenitic lattice with all rotations that define the symmetry of the parent phase ($\mathbf{U_i} = \mathbf{R_i U_1 R_i^T}$). Then depending on the symmetry of the martensitic phase some variants can coincide. Generally, the number of martensite variants N can be calculated by

$$N = \frac{\text{number of rotations in } \mathcal{P}_a}{\text{number of rotations in } \mathcal{P}_m} \tag{2.14}$$

In the case considered here this is $24/2 = 12$ since the monoclinic lattice has a two-fold rotational symmetry. In the literature either the set of directly rotated deformation gradients ($\mathbf{F_i} = \mathbf{R_i U_1}$) or the Bain Variants can be found. Again, it is emphasized that in this work the chosen Cartesian coordinate system is different from what is commonly used in literature (axes are interchanged). Since anisotropic elastic constants are used for the monoclinic phase it is important that these refer to the same coordinate system as the calculated strains, i.e. the Cartesian coordinate system of the cubic cell shown in Figure 2.4 a) obtained by rotating the X,Y,Z system of the tetragonal cell clockwise by 45° around the Y axis. All deformation gradients of different symmetry related variants calculated in this coordinate system can be found in Table 2.1. Note that the deformation gradient given in (2.13) is denoted (a) and that variants with opposite shear (denoted x and x' respectively) form a laminate as is elaborated in subsection 2.5.1. Analogously

all symmetry related Bain variants are given in Table 2.2. At this point it is worth clarifying one important thing also pointed out by Bhattacharya [21]: Although both material symmetry and frame indifference involve rotations, in material symmetry the rotation acts in the reference configuration and in frame-indifference the rotation acts in the deformed configuration. Therefore, it is not possible to rigidly rotate one variant to obtain another, i.e. $\mathbf{QU_1} \neq \mathbf{U_2}$ otherwise the polar decomposition theorem would not be unique. This also proves that the energy wells in Figure 2.3 c) are indeed unconnected.

Table 2.1.: Deformation gradients of all martensite variants represented in one and the same identical cubic basis. Variants of opposite shear form a twin laminate.

Variant	**elements of** $\underline{\underline{F}}_i$								
	$\boldsymbol{f}_{11}$	$\boldsymbol{f}_{12}$	$\boldsymbol{f}_{13}$	$\boldsymbol{f}_{21}$	$\boldsymbol{f}_{22}$	$\boldsymbol{f}_{23}$	$\boldsymbol{f}_{31}$	$\boldsymbol{f}_{32}$	$\boldsymbol{f}_{33}$
(a)	η	0	-ζ	-κ	ξ	κ	-ζ	0	η
(a')	η	0	-ζ	κ	ξ	-κ	-ζ	0	η
(b)	η	ζ	0	ζ	η	0	-κ	-κ	-κ
(b')	η	ζ	0	ζ	η	0	-κ	κ	κ
(c)	η	0	ζ	κ	ξ	κ	ζ	0	ξ
(c')	η	0	ζ	-κ	ξ	-κ	ζ	0	ξ
(d)	η	-ζ	0	-ζ	η	0	κ	-κ	ξ
(d')	η	-ζ	0	-ζ	η	0	-κ	κ	ξ
(e)	ξ	κ	-κ	0	η	-ζ	0	-ζ	η
(e')	-ξ	-κ	-κ	0	η	-ζ	0	-ζ	η
(f)	ξ	κ	κ	0	η	ζ	0	ζ	η
(f')	-ξ	-κ	κ	0	η	ζ	0	ζ	η

Elements of $\underline{\underline{F}}_i$: $\xi = 0.9612$, $\eta = 1.0215$, $\zeta = 0.0581$ and $\kappa = 0.1043$

Table 2.2.: Possible Bain-strain matrices for the representation in the cubic basis. The order is in analogy with Table 2.1 and the notation in accordance to [21]

$U_a = \begin{pmatrix} \alpha & -\epsilon & -\delta \\ -\epsilon & \gamma & \epsilon \\ -\delta & \epsilon & \alpha \end{pmatrix}$	$U_{a'} = \begin{pmatrix} \alpha & \epsilon & -\delta \\ \epsilon & \gamma & -\epsilon \\ -\delta & -\epsilon & \alpha \end{pmatrix}$
$U_b = \begin{pmatrix} \alpha & \delta & -\epsilon \\ \delta & \alpha & -\epsilon \\ -\epsilon & -\epsilon & \gamma \end{pmatrix}$	$U_{b'} = \begin{pmatrix} \alpha & \delta & \epsilon \\ \delta & \alpha & \epsilon \\ \epsilon & \epsilon & \gamma \end{pmatrix}$
$U_c = \begin{pmatrix} \alpha & \epsilon & \delta \\ \epsilon & \gamma & \epsilon \\ \delta & \epsilon & \alpha \end{pmatrix}$	$U_{c'} = \begin{pmatrix} \alpha & -\epsilon & \delta \\ -\epsilon & \gamma & -\epsilon \\ \delta & -\epsilon & \alpha \end{pmatrix}$
$U_d = \begin{pmatrix} \alpha & -\delta & \epsilon \\ -\delta & \alpha & -\epsilon \\ \epsilon & -\epsilon & \gamma \end{pmatrix}$	$U_{d'} = \begin{pmatrix} \alpha & -\delta & -\epsilon \\ -\delta & \alpha & \epsilon \\ -\epsilon & \epsilon & \gamma \end{pmatrix}$
$U_e = \begin{pmatrix} \gamma & \epsilon & -\epsilon \\ \epsilon & \alpha & -\delta \\ -\epsilon & -\delta & -\alpha \end{pmatrix}$	$U_{e'} = \begin{pmatrix} \gamma & -\epsilon & \epsilon \\ -\epsilon & \alpha & -\delta \\ \epsilon & -\delta & \alpha \end{pmatrix}$
$U_f = \begin{pmatrix} \gamma & \epsilon & \epsilon \\ \epsilon & \alpha & \delta \\ \epsilon & \delta & \alpha \end{pmatrix}$	$U_{f'} = \begin{pmatrix} \gamma & -\epsilon & -\epsilon \\ -\epsilon & \alpha & \delta \\ -\epsilon & \delta & \alpha \end{pmatrix}$

Elements of $\underline{\underline{U}}_i$: $\alpha = 1.0254$,$\epsilon = 0.0490$, $\delta = 0.0620$ and $\gamma = 0.9587$.

2.5. Twinning theory

This section intends to give an overview of the mathematical description of microstructures that can be formed with a combination of martensitic variants (twins). There are many related, but slightly different definitons of a twin [21, 47]. A graphical one shall be given here: A twin is a planar defect in a crystal where the lattice on the one side can be obtained either by a simple shear or a rotation of the lattice on the other side. Principally, twinning modes are determined as a consequence of the energy well structure elaborated in section 2.3. Practically, the Hadamard jump condition is used as a condition for a coherent interface, where more complex microstructures can be described with additional side constraints of further Hadamard jump conditions. The theoretical framework of this line of thought is quite mathematical since solutions for the compatibility of more complicated structures are rather restrictive and require a concise mathematical description. In the literature often the term non-linear theory can be found. It has to be mentioned that this could easily be misinterpreted as mathematical non-linearity, although the equations used to describe twinning without an external loading are linear. By contrast, the multi-well energy character of the evolving phase leads to a physically non-linear material behavior.

The governing equations for the compatibility of two variants (also twin plane or multiple well problem) are presented in subsection 2.5.1, where furthermore the considered laminate structure considered here (alternating parallel variant domains) is discussed along with a linearization, which is useful in some cases. Note that in the non-linear theory the set of all possible homogeneous boundary conditions is only known in the case of transformations involving two variants of martensite [10]. The description of other microstructures consisting of two variants like so called "Zig-zag" or "Crossing" twins can be found elsewhere [21].

In subsection 2.5.3 the formalism for describing habit plane structures by additional side constraints to the martensite variants is elaborated. In particular it will be shown that the non-linear theory reduces to the crystallographic theory of martensite for the habit plane between a twin laminate and austenite. Other habit plane structures like the one between two (or more) twin laminates, two twin laminates and the austenite (wedge-like microstructures), which are characteristic features observed in SMAs, are treated conceptually.

Generally, a main result of the non-linear theory is the conclusion that some common microstructures, like for example wedge-like microstructures, are only possible in shape memory alloys in a stress free manner with very special lattice parameters satisfying certain highly restrictive conditions [48].

2.5.1. Twin Plane Calculation

In this section we use the Hadamard jump condition (2.10) as twinning equation by setting the two deformation gradients $\mathbf{F}$ and $\mathbf{G}$ in (2.10) to two out of 12 possible Bain variants. Also, remembering the above definition of a twin its two variants must be related by a rotation $\mathbf{R}$, therefore:

$$\mathbf{R}\mathbf{U_j} - \mathbf{U_i} = \mathbf{a} \otimes \hat{\mathbf{n}} \tag{2.15}$$

The following algorithm can be used to find any possible twin described by a rotation $\mathbf{R}$, shear vector $\mathbf{a}$, and twin plane normal $\hat{\mathbf{n}}$ satisfying the twinning equation for a given pair of martensite variants $(\mathbf{U_i}, \mathbf{U_j})$. If (2.15) is post-multiplied by $\mathbf{U_i}$, then it follows that $\mathbf{R}\mathbf{U_j}\mathbf{U_i^{-1}} = \mathbf{I} + \mathbf{a} \otimes \mathbf{U_i^{-1}}\hat{\mathbf{n}}$. From this a symmetric matrix $\mathbf{C_t}$ is defined by multiplying each side with its transpose:

$$\mathbf{C_t} := \mathbf{U_i^{-1}}\mathbf{U_j^T}\mathbf{U_j}\mathbf{U_i^{-1}} = \mathbf{U_i^{-1}}\mathbf{U_j^2}\mathbf{U_i^{-1}} \overset{C_t \neq 0}{=} (\mathbf{I} + \mathbf{U_i^{-1}}\hat{\mathbf{n}} \otimes \mathbf{a})(\mathbf{I} + \mathbf{a} \otimes \mathbf{U_i^{-1}}\hat{\mathbf{n}}) \tag{2.16}$$

The six parameters that determine $\mathbf{a}$ and $\mathbf{n}$ can be obtained as proposed by Ball and James [18]: Necessary and sufficient conditions for a symmetric 3×3 matrix $\mathbf{C_t}$ with eigenvalues $0 \leq \lambda_1 \leq \lambda_2 \leq \lambda_3$ (ordered sequence) to be expressible in the form 2.16 for nonzero $\mathbf{U_i^{-1}}\hat{\mathbf{n}}, \mathbf{a}$ and $\mathbf{C_t}$ are given by:

$$\begin{aligned} \mathbf{a} &= \varrho \left(\sqrt{\frac{\lambda_3(1-\lambda_1)}{\lambda_3 - \lambda_1}}\hat{e}_1 + \chi\sqrt{\frac{\lambda_1(\lambda_3 - 1)}{\lambda_3 - \lambda_1}}\hat{e}_3 \right) \\ \hat{\mathbf{n}}\mathbf{U_i^{-1}} &= \varrho^{-1} \left(\frac{\overline{\chi}\sqrt{\lambda_3} - \sqrt{\lambda_1}}{\sqrt{\lambda_3 - \lambda_1}} \right) (-\overline{\chi}\sqrt{1-\lambda_1}\hat{e}_1 + \chi\sqrt{\lambda_3 - 1}\hat{e}_3) \end{aligned} \tag{2.17}$$

where ϱ is the norm of the calculated vector resulting in $\mid \hat{\mathbf{n}} \mid = 1$, $\hat{\mathbf{e}}_\mathbf{1}, \hat{\mathbf{e}}_\mathbf{3}$ are normalized eigenvectors of $\mathbf{C_t}$ corresponding to λ_1, λ_2 respectively and $\overline{\chi}, \chi$ can take the values ± 1. So $\hat{\mathbf{n}}$ is determined first, then ϱ, next $\mathbf{a}$ and finally $\mathbf{R}$ by reinserting into (2.15). A proof of (2.17) can be found in [18]. Subsequently, some remarks on solutions in dependence of the eigenvalues in (2.17) according to [18] are given:

(i) $\mathbf{I} + \mathbf{a} \cdot \mathbf{U_i^{-1}}\hat{\mathbf{n}} = \overline{\chi}\sqrt{\lambda_\mathbf{1}\lambda_\mathbf{3}}$

(ii) Consider solutions of (2.16) with $\mathbf{I} + \mathbf{a} \otimes \mathbf{U_i^{-1}}\hat{\mathbf{n}} \geq \mathbf{0}$. If $\lambda_1 < \lambda_2 = 1 < \lambda_3$, then there are two essentially distinct solutions related by a rotation $\mathbf{R}$ in the sense that $\mathbf{I} + \mathbf{a}^+ \otimes (\mathbf{U_i^{-1}}\hat{\mathbf{n}})^+ = \mathbf{R}(\mathbf{I} + \mathbf{a}^- \otimes (\mathbf{U_i^{-1}}\hat{\mathbf{n}})^-)$; this follows from the polar decomposition theorem in the case $\lambda_1 > 0$ and by an explicit calculation if $\lambda_1 = 0$. If λ_1 or λ_3 equals 1, there is only one solution. If λ_1 and λ_3 are both equal to 1, there is no solution.

Note that physically the eigenvalues of the matrix $\mathbf{C_t}$ describe the stretches of one side of the twin plane relative to the other. Coherency of the two sides after each one has deformed requires that one eigenvalue is equal to one. This is only possible if one of the other two is less than 1 and the other greater than 1.

A comprehensive application of the above results for NiTi can be found in [49] for example. The above equations are comprehensive and yield all possible coherent variant pairings (twins). In other words, an isolated single crystal composed of only one variant would have the same free energy as the same crystal composed of two variants fulfilling the twinning equation. In the bulk however, there are additional constrains that the crystal has to fulfill in order to minimize the free energy. Obviously, a minimal misfit to the matrix and consequently a minimal misfit strain energy can be achieved if the martensite forms a very fine mixture of two variants called a twin laminate or fine twins. These fine twins are described as weakly converging sequences fulfilling that each deformation is continuous (note that the deformation gradient not necessarily is continuous) and that the deformation gradients do not converge locally but on average and also do not become larger. From this point of view fine twins exhibit scale invariant characteristics similarly to fractals. Mathematically, the minimization can be performed by a variational caluclus, where the functional is of the general form:

$$I_T(y) = \int_\Omega g(\mathbf{F}, T)dx \qquad (\ + \text{loading energy}\ ) \tag{2.18}$$

Neglecting external loading this variational approach principally yields an infinitely small twin variant width. Taking loading into account, this generally leads to a difficult nonlinear elasticity problem [22]. Comprehensive reviews on variational approaches concerning this topic can be found in [50, 51].

It is worth mentioning that as for all geometric problems there exists a linearized version of the above problem, meaning physically nonlinear but geometrically linear. In the linear theory the multiplicative polar decomposition for example is replaced by an additive decomposition $\mathbf{F} = \mathbf{RU} \Rightarrow \mathbf{H} = \mathbf{E} + \mathbf{W}$ where $\mathbf{H} = \nabla\mathbf{u}(\mathbf{x}) = \mathbf{F} - \mathbf{I}$ ($u(x) = y(x) - x$) is called the displacement gradient, $\mathbf{E} = \mathbf{1/2}(\mathbf{H} + \mathbf{H^T})$ the infinitesimal strain matrix and $\mathbf{W} = \mathbf{1/2}(\mathbf{H} - \mathbf{H^T})$ the infinitesimal rotation matrix. As for all linearizations it must be pointed out that this is only a good approximation as long as changes (here $\mathbf{H}$ or $\mathbf{E}$ and $\mathbf{W}$) are small. The wells are then defined by linear strain matrices and the twinning equation becomes the strain compatibility equation, since on many occasions only the strains need to be considered.

$$\mathbf{E_H} - \mathbf{E_K} = \frac{\mathbf{1}}{\mathbf{2}}(\mathbf{a} \otimes \hat{\mathbf{n}} + \hat{\mathbf{n}} \otimes \mathbf{a}) \tag{2.19}$$

Also for this equation there are procedures that lead to a solution as well as interpretations of the solutions. A modified version of the linear theory is the piecewise linear theory, whose main assumption is that the wells of the free energy density are piecewise approximated quadratic functions. This enables to describe any material as a composite of linear elastic materials with different residual strains. For details on the computation-

ally interesting piecewise linear theory it is referred to [52].
Finally, again it has to be pointed out that all the above theories globally only apply to single crystals and are only useful in a polycrystal for a local minimization inside a single grain. A mathematical continuation of compatibility in polycrystals can be found in [21].

2.5.2. Classification of Twins

From remark (ii) to (2.17) it can be seen that the solutions come in pairs. The two different twins obtained from a solution are called reciprocal twins. There are two other popular results that are easier to use, but which are (i) not comprehensive and (ii) are only applicable if the martensite variants are related through a rotation in the point group of austenite. The first one is called "Mallards Law" [21, 22] and works for variants related by a 180° rotation in the nonlinear theory. It is useful since most twins in the martensite are of this type and because it allows a classification of twins to type I (twin plane is a plane of symmetry) and type II (shearing direction is a direction of symmetry) respectively. There are twins that fulfill both conditions and are consequently termed compound twins. Type I and II twins are often referred to as conventional twins. In nanograined NiTi mainly those are found. The second result is due to Forclaz and provides a quick check if two variants of the above type fulfill the twinning equation. It states that this is the case for two martensite variants $\mathbf{U_i}, \mathbf{U_j}$ if $det(\mathbf{U_i} - \mathbf{U_j}) = 0$.
Another classification divides twins into generic and non generic ones. Generic twins are transformation twins that can form for any lattice parameters in the martensite and are thus purely “symmetry-driven”. By contrast, the existence of non-generic twins is restricted to particular lattice parameters. Or from a mathematical point of view: For fixed $\mathcal{P}^a$ and $\mathcal{P}^m$ the eigenvectors of the Bain strain $\mathbf{U}$ are unique, whereas the eigenvalues depend on the lattice parameters of the austenite and martensite phase [47].

2.5.3. Calculation of Habit Plane Structures

In this section the mathematical construction of habit plane structures in the non-linear theory is summarized. The two main assumptions are: (i) Twin laminates are described as weakly converging sequences (see subsection 2.5.1). (ii) As a consequence of energy minimization, the gradients in this sequences only take values in the energy wells. In principle, based on this assumptions criteria are constructed under which the above equations for a coherent interface can be applied. The easiest case of a habit plane between the austenite and a single martensite variant is not discussed here since it is strongly, non-generic. A

commonly observed and historically the most important habit plane structure is between austenite, seen as a domain of constant deformation gradient **I** and finely twinned martensite as illustrated in Figure 2.5. In order to make the deformation between the laminate

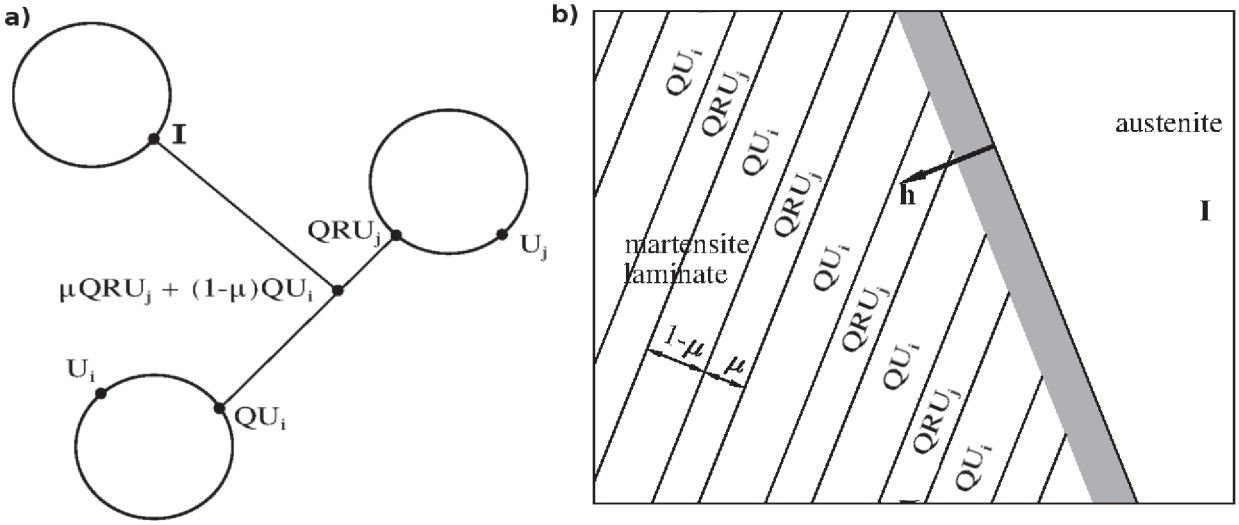

Figure 2.5.: Schematic representations of an austenite-martensite interface. a) Energy well representation and b) the microstructure arrangement. Austenite, with the deformation gradient I, is separated from martensite by the habit plane with the normal $\hat{\mathbf{h}}$. The grey space indicates a transition zone since the interface is not sharp, however the deformation is continuous across it.

and the austenite continuous the laminate is assumed to be very fine. Let the volume fraction of the variant with deformation gradient $\mathbf{QRU_j}$ be $\mu \in (0,1)$, then the average deformation gradient of the laminate is:

$$\mathbf{F}_\mu = \mathbf{Q}[\mu \mathbf{RU_j} + (\mathbf{1} - \mu)\mathbf{U_i}] \tag{2.20}$$

Since the deformations on both sides of the habit plane must be compatible:

$$\mathbf{F}_\mu - \mathbf{I} = \mathbf{b} \otimes \hat{\mathbf{h}} \tag{2.21}$$

where the vector $\hat{\mathbf{h}}$ is the habit plane normal and the vector $\mathbf{b} \in R^3$ is called shape strain. Additionally, for the laminate the twinning equation (2.15) applies, therefore the compatibility condition comprises a system of two equations. The austenite-twinned martensite interface was first described by the crystallographic theory. Therefore, at this point the equivalence of the non-linear and crystallographic theory is by inserting the twinning equation (2.15) into the expression for $\mathbf{RU_j}$ in (2.21).

$$\mathbf{Q}[\mu \mathbf{a} \otimes \hat{\mathbf{n}} + \mu \mathbf{U_i} + \mathbf{U_i} - \mu \mathbf{U_i}] = \underbrace{\mathbf{Q}}_{\mathbf{R}} \underbrace{[\mathbf{I} + \mu \mathbf{a} \otimes (\mathbf{U_i^{-1}} \hat{\mathbf{n}})]}_{\mathbf{P_2}} \underbrace{\mathbf{U_i}}_{\mathbf{B}} = \underbrace{\mathbf{I} + \mathbf{b} \otimes \hat{\mathbf{h}}}_{\mathbf{P_1}} \tag{2.22}$$

Combining the Chauchy Born hypothesis, the twinning equation and a result from matrix algebra it can be shown [21] that the matrix $(\mathbf{I} + \mu \mathbf{a} \otimes (\mathbf{U_i^{-1}} \hat{\mathbf{n}})$ is a simple shear. There-

fore, the non-linear theory in this case is equal to the crystallographic theory outlined in subsection 1.1.1. Conceptually, the advantage of Equation 2.21 is that F_μ describes the macro-scale deformation of the laminate.
From the theory of Ball and James it is clear, that in order to solve the coherency condition for the eigenvalues of any $\mathbf{C}$ matrix, describing the deformation of one side relatively to the other, the following has to hold true: $\lambda_1 < 1, \lambda_2 = 1, \lambda_3 > 1$. They also derive an existence theorem ensuring that this condition is fulfilled and Equation 2.21 has a solution [18]. From the left hand side of (2.22) it is easy to check that the $\mathbf{C}$ matrix for Equation 2.21, describing the deformations around the habit plane, has the form:

$$\mathbf{C_h} = (\mathbf{U_i} + \mu\hat{\mathbf{n}} \otimes \mathbf{a})(\mathbf{U_i} + \mu\mathbf{a} \otimes \hat{\mathbf{n}}) = (\mathbf{I} + \hat{\mathbf{h}} \otimes \mathbf{b})(\mathbf{I} + \mathbf{b} \otimes \hat{\mathbf{h}}) \tag{2.23}$$

Then solutions for $\mathbf{b}$ and $\hat{\mathbf{h}}$ can be found using (2.17) where $\mathbf{b} = \mathbf{a}$ and $\hat{\mathbf{h}} = \mathbf{U_i^{-1}}\hat{\mathbf{n}}$. Note that for a given martensite laminate, specified by twin related variants $\mathbf{U_i}$ and $\mathbf{U_j}$ and a corresponding twinning shear $\mathbf{a}$ and normal $\hat{\mathbf{n}}$, there are up to four distinct solutions for the habit plane equation. It is worth mentioning that these interfaces are irrational or high indexed planes that are also experimentally observed. This is especially important since a mere experimental (e.g. X-Ray) detection of high indexed planes would be highly controversial because of measurement inaccuracies. Overall, there are 192 possible distinct austenite-martensite interfaces in NiTi. This is especially interesting since 192 is a multiple of 24, the number of symmetry rotations in the parent cubic phase.

At this point it should be clear that other habit plane structures can similarly be constructed. For instance consider a twin within a twin. For this microstructure one obtains three equations. One for each twin laminate, respectively, and one for that describing their connection. This line of thought can be carried on to a twin within a twin within a twin where the system of coherency equations for an overall compatibility already involves 8 equations. Another, more popular and above all thermodynamically reasonable example is the so called wedge like microstructure. This microstructure is basically a twin within a twin, which sharpens on one side to a corner and where each twin laminate additionally forms an austenite-martensite interface, leading to a system of 5 equations.

Note that the constructive methods above are inherently limited to answering the question if a material can generally form a self-accommodating microstructure. To quickly rule out impossible microstructures so called average compatibility conditons or minor relations (mathematically also referred to as cofactor conditions) have been developed [53].

2.6. Energy Contributions

Thermodynamically, phase transformations are described by the free energies (here denoted as G) of the participating phases. Note that by convention extensive variables are denoted with capital letters, whereas intensive variables are denoted with lowercase letters. In this work we study a volume preserving (dV=0) solid state transformation (closed system), where the change in free energy without a transformation is simply

$$dG = -SdT \tag{2.24}$$

where S is the entropy. Commonly, the free energies of the phases taking part in the transformation are plotted against temperature as linear functions. It is emphasized that this is only a reasonable approximation in the vicinity of the free energy equilibrium of the two phases since the free energy function actually is curved. The intersection of the two curves marks the equilibrium temperature T_0 where $G_a = G_m$. Above ($T > T_0$) and below that point ($T < T_0$) the phase with the lower free energy is stable from a pure equilibrium-thermodynamic point of view. In practice, the forward transformation from austenite to martensite does not take place directly under T_0, but only at a certain under-cooling $\Delta T = T_0 - M_s$ as illustrated in Figure 2.6. The amount of undercooling is proportional to a free energy barrier, i.e all additional emerging energies in the wake of the transformation, here denoted as E_b. Commonly, the free energy barrier is expressed in the form that the negative free energy change resulting from the formation of a given volume of a more stable phase is opposed by a positive free energy change due to the creation of an interface between the initial and the new phase as well as other positive contributions such as the strain-energies as a consequence of the transformation in solid matter. In the literature this transformation barrier is commonly termed nucleation energy. At this point it is mentioned that a phase is called metastable if thermodynamically another phase should be stable but is not due to a transformation barrier. In the model presented here the initial state is considered as stress free austenite. A time dependent homogeneous temperature field is assumed in the whole model. Thus, the energy density $-\int sdT$ is equal in all grains of the same phase. To evaluate an energy minimizing configuration also all dragging forces must be assigned specific values $E_b \Rightarrow e_b$. Upon cooling the austenite phase below T_0, g_m becomes lower than g_a and a specific driving force Δg, acts on the austenite to change it to the thermodynamically more stable martensite. The moment the first martensite forms, the amount of driving force is equal to the energy barrier. In this work the energy barrier is constituted of several interface-energies (IEs) as well as elastic strain energy as a consequence of the normal eigenstrain of each laminate (U_v) and a dissipative work of friction for the transformation (F_c) [54] . Contributions to the IE term are the twin-plane IE (Γ_{IN}), the energy of the austenite-martensite interface (Σ_S)

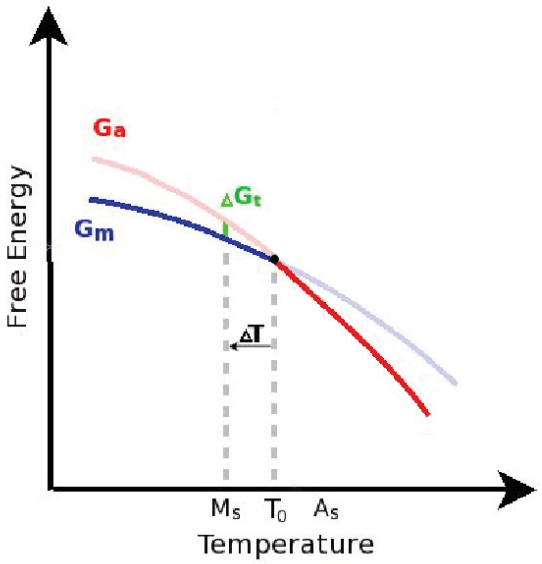

Figure 2.6.: Free energy curves of austenite (G_a) and martensite (G_m) as a function of temperature. T_0 marks the thermodynamical phase equilibrium. However, in practice an an additional undercooling is necessary in order to overcome the transformation barrier

and the twin surface-energy (Γ_{TW}) [54]. The first two terms are discussed in chapter 4 and the last one in section 3.1. The governing equation of the specific free energy function for each grain is fully written as:

$$0 = \underbrace{-\Delta g}_{Driving force} + \underbrace{(\Gamma_{IN} + \Gamma_{TW} + \Sigma_S)\frac{A_i}{V_i} + F_c + \frac{U_{vi}}{V_i}}_{Energy\ barrier} \tag{2.25}$$

The contribution of IEs to the energy-barrier strongly depends on the coherency of emerging interfaces, but is comparatively low in martensitic transformations. However, it is emphasized that IE contributions increase with decreasing grain-size since the surface to volume ratio increases, hence IEs have to be considered at the nano-scale investigated here. Note that IEs basically emerge due to misfits in the atomic structure between two phases. Therefore considering nano-structured materials these misfit energies can be approximately determined as misfit strain-energies at a near atomic scale if the morphology and the materials elastic constants are well known, see section 3.1.

3. Model

In this chapter a three dimensional multi-variant model is presented. The model developed here simulates a purely thermally activated martensitic transformation in a nanostructured NiTi microstructure without any preferred texture. No external loadings are applied, so no variant selection due to macroscopic stresses or detwinning is expected. In this work it is also intended to investigate the influence of the grain morphology on the transformation. To this end a regular and a random grain morphology, respectively, with different boundary conditions are developed. The evolving strain-energies are calculated by the finite element method and IEs are accounted for semi analytically. After some manual preprocessing the whole simulation proceeds fully automated only requiring the Abaqus standard finite element solver and a working python setup. Computational aspects of this automation are discussed by pointing out programming techniques of the scripting language Python and its implementation for the finite element software Abaqus. Finally, considerations regarding the transformation algorithm and the accompanying meshing issues as well as the pre- and postprocessing are presented.

3.1. Interface Energy Model

The model for the interface energies (IE) in this work follows [54] and is subsequently reviewed. First and foremost all interface energies of a specific grain are calculated for a hypothetical spherical grain with the same volume. After the calculation of a specific interface energy in $[J/m^2]$ it is related to the surface of the volume-equivalent grain and not to the original surface although it would be possible to implement a quick-hull algorithm to get the exact surface of a certain grain. However, since the model in [54] is derived for a spherical grain equivalent sizes are considered. Note that since a sphere has the largest volume to surface ratio, this approach can be considered a lower boundary of the interface energy. Also note that although the strain energy dominates the energy barrier, interface energies should particularly be considered. First in order to obtain more exact results and second and more importantly because they exhibit a typical grain size dependent scaling, which explains the suppression of the martensitic transformation at low grain sizes. In the calculation of the interface energies isotropic elastic constants are considered also for martensite, which are deliberately set higher ($E_m = 150, E_A = 75$ GPa) than the ones used for the strain energy calculations given in section 4.3 for two reasons. (i) To partially compensate the smaller surface of the equivalent volume sphere. (ii) To account for martensite neighbors since in [54] only one grain in an austenitic matrix is considered. An overview of important IE contributions in this model is visualized in Figure 3.1. The

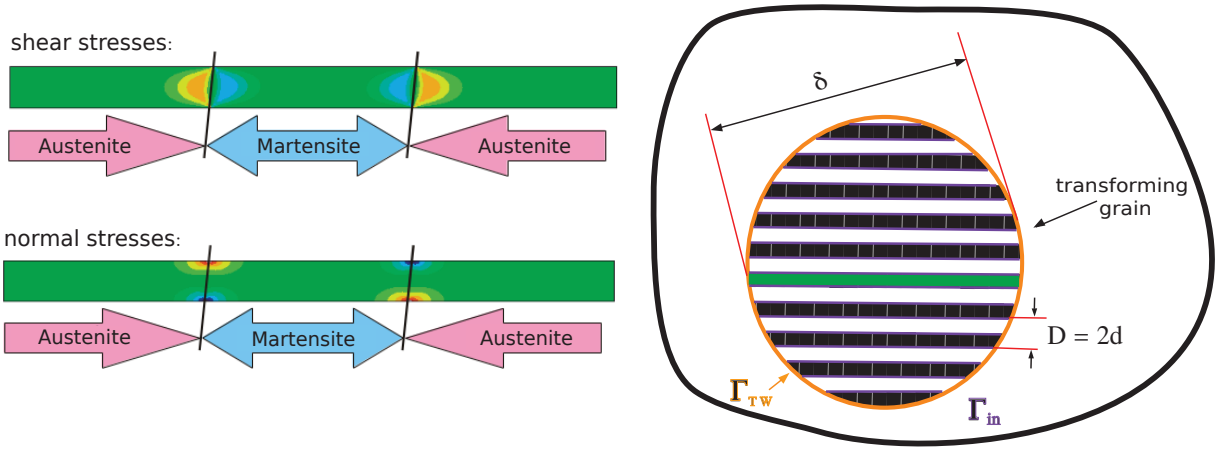

Figure 3.1.: Illustration of IEs: Γ_{TW} is the concentrated strain energy caused by the shear of the martensite variants at the interface to the austenite matrix. Note from the left hand side that the shear stresses are symmetric and the normal stresses are antisymmetric w.r.t to the twin axes. Γ_{in} is the low IE at the twin plane due to an atomic mismatch.

first IE is the surface strain energy Γ_{TW} $[J/m^2]$. Since the strain energy caused by the opposite shear eigenstrains of the martensite variants in the twin laminate is concentrated at the ends of the twinned domain near the matrix, it is reasonable to relate the strain energy part resulting from a pure shear to the surface area of the twinned domain. It can

be shown that the following relation applies to an infinite twin laminate [54]:

$$\Gamma_{TW} = f_{TW}(C)\alpha_{TW}G_A\gamma_T^2 2d \tag{3.1}$$

where $\gamma_T = 2\sqrt{\epsilon_{13}^2 + \epsilon_{23}^2}$, G_A is the shear modulus of austenite in GPa, which is related to the Young's modulus of austenite by $G_A = E_A/(2(1+\nu))$, d is the thickness of one martensite variant in the laminate. Note that in [54] the two variants lamellae are termed double-twin with the width D. The factor $f_{TW}(C)$ accounts for the elastic contrast C, which is the ratio of the isotropic modulus in the austenitic matrix to that in the transformed inclusion: $C = E_A/E_M = G_A/G_M$. α_{TW} depends on the Poisson's ratio and is for an infinite stack of double twins [54]: $\alpha_{TW} = 0.856(1+\nu)/8\pi$. f_{TW} is determined numerically using FEM from a single lamella inclusion modeled with plane stress elements and appropriate symmetry conditions on its boundary, obviously sufficient for the case of a periodic sequence of double twins, by setting Γ_{TW} $[J/m^2]$ equal to the inclusion's strain energy per thickness of the FEM model U_I $[J/m^3]$ / t $[m]$.

$$f_{TW}(C) = \frac{U_I}{\alpha_{TW}\, G_A\, \gamma_T^2\, 2\, d\, t} \tag{3.2}$$

Note that since U_I depends on d, f_{TW} is constant for an arbitrarily chosen d. An additional term on the grain boundary Σ_S is considered accounting for the difference of the non-mechanical IE of the initial grain boundary energy prior to transformation Σ_a and the interface energy after transformation Σ_m ($\Sigma_S = \Sigma_m - \Sigma_a$).
The twin-boundary energy Γ_{in} $[J/m^2]$ is a size which relates the IE obtained from the area of all twin-planes in the laminate ($A_{tot}(\delta)$) multiplied by a specific twin-boundary energy $\Sigma_{TW}[J/m^2]$ to the equivalent sphere's surface $O_s = \delta^2\pi$ ($m^2 \cdot J/m^2 \cdot 1/m^2 = J/m^2$). After some algebra [54]

$$\Gamma_{in} = \frac{A_{tot}(\delta)}{\delta^2\pi}\Sigma_{TW} \approx \frac{\delta}{6d}\Sigma_{TW} \tag{3.3}$$

As can be seen from 3.1 and 3.3 the IEs show different scaling behavior with respect to the lamella width d: $\Gamma_{TW} \propto d^{-1}$ and $\Gamma_{in} \propto d$. Obviously, from an energetic point of view an optimum lamella width d_{opt} can be obtained by a minimization of the energy density.

$$\begin{aligned} \frac{\partial}{\partial d}\left(\Gamma_{TW}\frac{O_s}{V_s} + \Gamma_{in}\frac{O_s}{V_s}\right) &= \frac{12}{\delta}\alpha_{TW}G_A\gamma^2 f_{TW}(C)d - \frac{\Sigma_{TW}}{d} = 0 \\ \Rightarrow d_{opt}(\delta) &= \sqrt{\left(\frac{\Sigma_{TW}}{12 f_{TW}(C)\alpha_{TW}G_A\gamma_T^2}\,\delta\right)} \end{aligned} \tag{3.4}$$

d_{opt} is in good agreement with the experimental evidence. For $d = d_{opt}$ $\Gamma_{TW} = \Gamma_{in}$ and the sum of both increases proportional to $\delta^{-1/2}$, thus fulfilling the criterion of a larger energy barrier to the transformation, as stated introductory.

3.2. Artificial Microstructure Models

3.2.1. Random Voronoi Tesselation

The first investigated artificial microstructure is generated by the polycrystal generator "Neper" [55]. It consists of three dimensional Voronoi cells around randomly distributed points. The so called Voronoi diagram, i.e. the graph of the Voronoi cells, is generated by a well known algorithm. Note that the Voronoi diagram is dual to the Delaunay trianglulation, a common algorithm for the generation of a mesh consisting of triangular faces like that of the tetrahedral element mesh used here. The subsequent mesh generation additionally uses the gmsh package [56] within Neper. Neper has a lot of useful features shortly outlined subsequently:

1. Specification of the point distribution for the germs of the Voronoi cells. In work presented here a Poisson-Voronoi tesselation is used resulting from randomly chosen positions of the germs. Alternatively Neper can generate a wide variety of microstructures such as so called Hardcore Voronoi tesselations, Centroidal Voronoi tesselations and Laguerre Voronoi tesselations [57].

2. Creation of free or structured mesh types for linear and quadratic shape functions

3. Statistics on the generated structures

4. Definition of characteristic mesh lengths for certain regions, including the definition of a biased mesh.

5. An option to select grains that lie a certain number of grains (x) under the surface. This means that from such a grain the shortest path to the surface crosses at least x grains.

The last two points were combined to generate a refined mesh in the region of the transforming grain cluster and a coarser mesh for the region far away from it. This is done to keep the overall node and element number low enough so that the calculation time for one transformation is reasonable, since many such calculations have to be performed to find the energy minimum for each transformation state as is explained in subsection 3.5.2. The matrix of outer grains, which is not evaluated, is modeled in a self-consistent framework described in section 3.3. Figure 3.2 shows the full cubic matrix, a view cut of it and the inner evaluated grain-cluster constituting the representative volume element (RVE). The average grain size of the generated grain-cluster is $\approx$ 80nm equivalent-volume sphere's diameter. The grain size frequency distribution is approximately normal-distributed. The

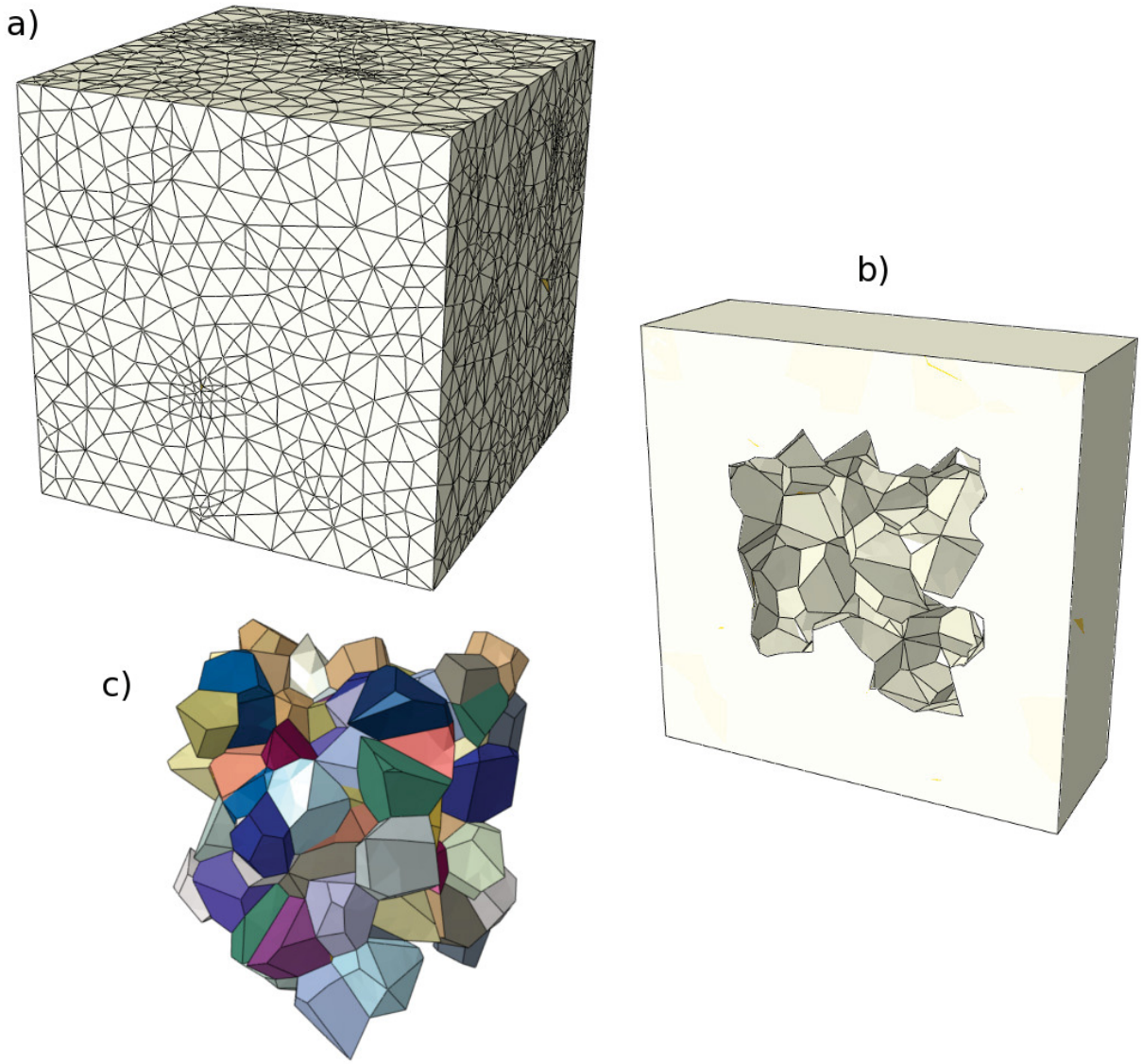

Figure 3.2.: a) cubic matrix with coarse outer mesh. b) view cut of a). c) extracted representative volume element (RVE) of clustered grains with a finer mesh.

volume of the largest grain is about 8 times larger than that of the shortest grain, resulting in an equivalent sphere diameter, which is only twice as large. Therefore, the grain size frequency distribution spans an interval from about 50 to 100 nm equivalent sphere's diameter. The nanograined NiTi microstructure reported in [35] reveals a broader grain size distribution in an interval >100 nm starting at the small grain size of 15 nm. Therefore, the artificial microstructure misses especially very small grains observed in a real microstructure. Nevertheless, the generated microstructure morphology is quite realistic and since the transformation is reported to be dominated by larger grains [35] the model seems justified.

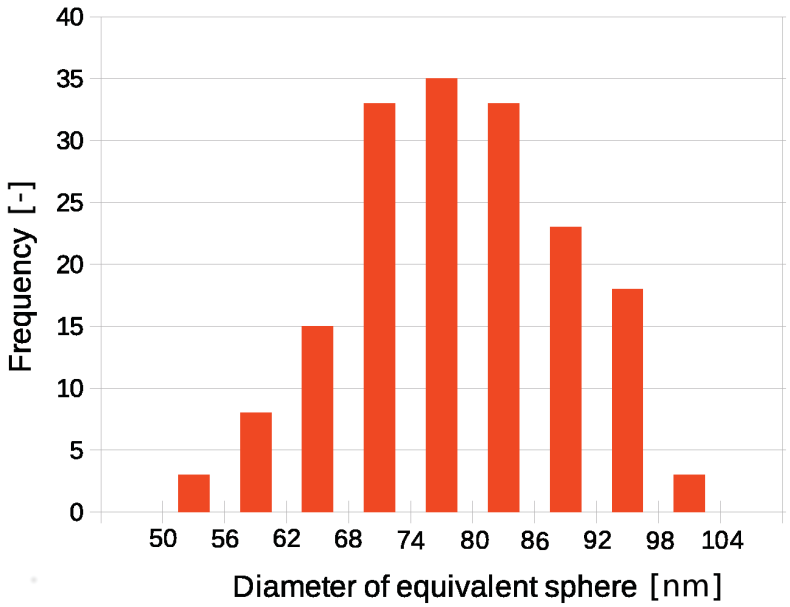

Figure 3.3.: Grain size frequency distribution of the random Voronoi microstructure. Note that this distribution is narrower than in a real microstructure like the one reported in [35]

3.2.2. Regular Tessellation of Truncated Octahedra

Beside the random microstructure also a regular one is created due to two reasons:
(i) For a regular tessellation periodic boundary conditions (PBCs) can be developed. PBCs are desirable to accurately predict the strain energy development inside a bulk. Defining PBCs is not possible with Neper, at least not for a tetrahedral mesh. For details on the PBCs see section C.2.
(ii) IEs become relevant for the selection of the transforming grain since they are equal for all grains. Hence the free energy barrier is purely controlled by the strain energy.
The goal is to fill the three dimensional Euclidean space with one kind of a polyhedron. Among all convex polyhedra (only confined by regular polygons) only a few are space-filling e.g. a cube, three- and six-sided prisms, truncated octahedra and the rhombic dodecahedron. These polyhedra are congruent to their symmetric copies and are therefore said to be cell-transitive or isochoric, as already explained in 2.1, a property which makes them space-filling polyhedra. In geometry, such a space filling or close packing of three dimensional solids leaving no gaps, with flat faces and straight edges (polyhedral), is also called a honeycomb. It is furthermore called a uniform honeycomb if it is composed of only one uniform polyhedral cell. In this model a RVE consisting of truncated octahedra is used. The truncated octahedron is the Voronoi diagram of the bcc cubic lattice. Beside the truncated octahedron honeycomb the rhombic dodecahedron honeycomb is a popular example since it represents the Voronoi diagram of the face-centered cubic sphere-packing, which is the densest possible packing of equal spheres in three dimensions. A space filling cell of truncated octahedra is illustrated in Figure 3.4 consisting of two linear tesselations

of truncated octahedra interlocked into each other, similar to the simple-cubic sub-lattices of Na and Cl, respectively, in a NaCl lattice.

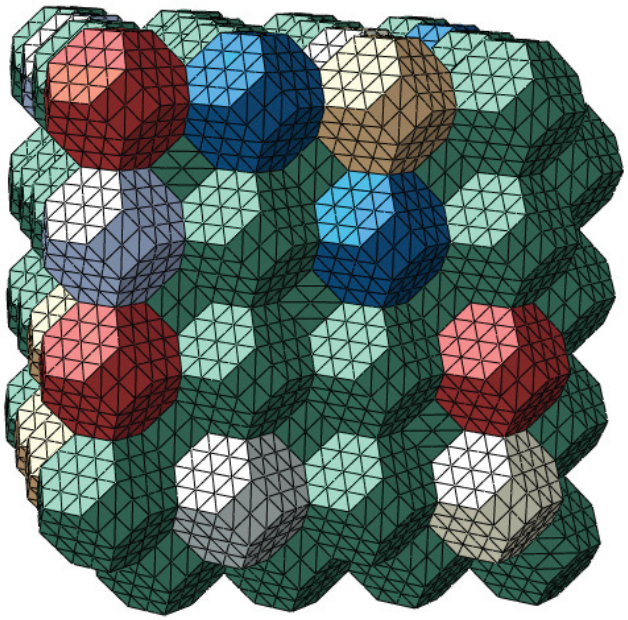

Figure 3.4.: Space filling RVE of truncated octahedra representing grains of a regular microstructure

The characteristic length of the tesselated truncated octahedron is calculated again by means of an equivalent-volume sphere with the characteristic average grain diameter of 80 nm as for the random Voronoi microstructure. Note that this is nearly equal to the average of its midsphere and inscribed sphere's radius due to the highly symmetrical structure of the truncated octahedron.

3.3. Self-Consistent Matrix

At a first glance a belt of transforming grains around the evaluated cluster of grains seems to be a suitable boundary assumption for the random Voronoi microstructure. However it is not clear which and when a grain in the boundary belt should transform. Also adding the boundary belt to the calculation of the overall energy minimum is not advisable since only a limited inner cluster of grains can be evaluated within reasonable computation times. Therefore, for this kind of RVE a self-consistent embedding in analogy to classical micromechanics is used. More precisely, the matrix surrounding the grain-cluster is given the average properties of the RVE. This means that from the anisotropic elastic constants related to each grain's local random orientation the grain volume weighted average of the

local elastic tensor is calculated for the whole RVE.

$$\underline{\underline{C}}^{ave} = \frac{1}{V_{ges}} \sum_{i=1}^{N} V_i \underline{\underline{C}}_i \tag{3.5}$$

The terms $\underline{\underline{C}}_i$ in (3.5) are the elastic tensors of each grain respectively expressed in the global coordinate system. The general equation for the rotatin of a fourth order tensor in index notation using Einsteins summation convention is:

$$C_{ijkl} = T_{im} T_{jn} T_{kp} T_{lq} C_{mnpq} \tag{3.6}$$

where the matrix $\underline{\underline{T}}$ maps the local coordinate system to the global one. $\underline{\underline{T}}$ generally can be obtained by describing the initial coordinate system in the new one. Then $\underline{\underline{T}}$ is the matrix of prefactors (also known as directional cosines). Hereby, the definition of the directional cosine via the dot product should be recalled. In this model Eulerian angles are used to define $\underline{\underline{T}}$. Generally, Eulerian angles are used to correlate two orthogonal coordinate systems with a common origin, where the transformation from one coordinate system into the other is achieved by a series of plane rotations. The rotations are performed about coordinate system axes generated by the previous rotation step. This transformation however is not unique. There are exactly six different ways of selecting the order of the three different axes. Here in the order was chosen to be z_1, y_2, z_3 resulting in rotation matrices of the general form:

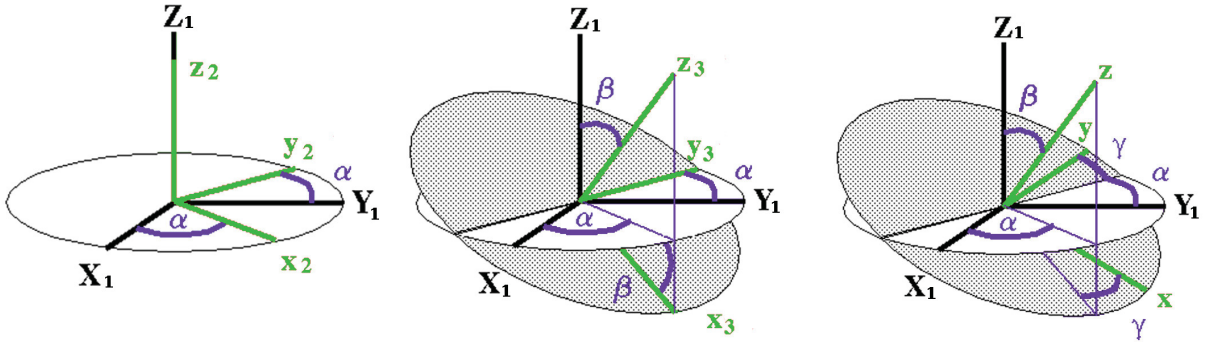

Figure 3.5.: Visualization of the three plane rotations and their corresponding Eulerian angles for the here chosen order of plane rotations describing the coordinate transformation.

$$\underline{\underline{T}} = R_{z1}(\alpha) R_{y2}(\beta) R_{z3}(\gamma) =$$

$$= \begin{pmatrix} cos(\alpha) & sin(\alpha) & 0 \\ -sin(\alpha) & cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} cos(\beta) & 0 & -sin(\beta) \\ 0 & 1 & 0 \\ sin(\beta) & 0 & cos(\beta) \end{pmatrix} \begin{pmatrix} cos(\gamma) & sin(\gamma) & 0 \\ -sin(\gamma) & cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} cos(\alpha)cos(\beta)cos(\gamma) - sin(\alpha)sin(\gamma) & sin(\alpha)cos(\beta)cos(\gamma) + cos(\alpha)sin(\gamma) & -sin(\beta)cos(\gamma) \\ cos(\alpha)cos(\beta)sin(\gamma) - sin(\alpha)cos(\gamma) & -sin(\alpha)cos(\beta)sin(\gamma) + cos(\alpha)cos(\gamma) & sin(\beta)sin(\gamma) \\ cos(\alpha)sin(\beta) & sin(\alpha)sin(\beta) & cos(\beta) \end{pmatrix}$$

Recalling Einsteins summation convention, from (3.6) it is straightforward to program the rotation element wise (see section E.5). In the simulation the self consistent matrix properties are recalculated in each increment. Although no stresses are introduced by the self consistent matrix into the grain-cluster RVE, the difference of overall generated stresses in the RVE due to hardening in the matrix nevertheless is significant because of the high difference of stiffness between martensite and austenite.

3.4. Comparison of Microstructures and Boundary Conditions

The regular morphology of truncated octahedra actually simplifies the calculation procedure, because the evaluated properties do not have to be weighted by the volume since it is equal for all cells. On the other hand random Voronoi cells are geometrically more realistic than the regular tesselation. However, the evaluation is not as straight forward because all energy densities have to be weighted by their individual grains volumes. As stated above PBCs are desirable, since the self-consistent matrix does not transfer the stresses realistically from outside the RVE. However, these stresses are significant for the in-bulk behavior. Note that this is especially important in this work because of the high surface to volume ratio as a consequence of the very small RVE size. Note that the mesh of the periodic RVE and the here specified boundary equations can be used in other models as well. On the other hand, it is more difficult to generate a periodic mesh and in further consequence apply PBCs to its boundaries. The advantage of the self-consistent scheme is that it can be applied to any mesh, which simplifies the preprocessing of the geometry. Both models, PBC and embedded RVE, represent boundaries for the energy calculations bracketing the energy found in realistic microstructures.

3.5. Computational Aspects

3.5.1. Python Tools and Abaqus Phyton

Scripting is a powerful tool that enables to combine the functionality of the Graphical User Interface (GUI) of Abaqus called “CAE” and the power of the programming language Python [58] [59]. Python is a general-purpose, high-level programming language, emphasizing code readability. Therefore it is the language of choice for scripting especially for non professional programmers. In Abaqus, scripting is used mainly to perform a repetitive task to save time. Using the Abaqus GUI and an editor for Python a script is developed the following way: First a .mdb file (model database) must be created using CAE. Every operation carried out in CAE is written to a file with the ending .jnl (journal) in the Python script language at every file-save in CAE. This file then is used to create a Python script with the file extension .py. A Python script file can be run directly in CAE via “FILE” - “Run script” or on the command line by typing “abaqus python scriptname.py”. Also the automation of processing data from the generated *.odb (output database) file can be achieved using a script. However, in the case of reading specific data the GUI is not that helpful any more because the user has to familiarize with the highly hierarchical object structure (called Abaqus object model). Abaqus .odb and also .mdb files are organized as is briefly outlined in section B.2. For using the Abaqus scripting interface to access an output database, it is necessary to understand how an Abaqus analysis writes data to the output database as well as the difference between field data, history data, and model data. Also there are several levels at which output data can be written ranging from the integration point level, via an element and element set level, to the level of the whole model. However, some properties are only available at an integration point level. Therefore it is necessary to make weighted averages if these properties are requested at a higher level as is done in section E.3. In the “Scripting Reference Manual” of the Abaqus documentation [60] an overview of all objects can be found. In this work the Abaqus Python interface is only used for an automatic evaluation of the results, so that no model data is manipulated using the Abaqus Python interface. Python is used to modify model data by automating the creation of new slightly modified input-files depending on the results of previous calculations. The manipulation of input files via scripts is preferred in this case because the transformation algorithm requires a huge amount of similar jobs to be calculated, which is achieved more efficiently by parallel computing outside CAE only invoking the Abaqus solver. Related commands for input files are found in the “Abaqus Keyword Reference Manual” of the Abaqus documentation [60].

3.5.2. Transformation Algorithm

The key features of the RVE model can be briefly summarized as follows:

- The polycrystal is approximated by a RVE of many polyhedra constituting the entire polycrystal.

- The grains are small enough so that only a single laminate martensite morphology will appear in the grains upon transformation. Since experimentally only six laminates are observed only these are considered within this model, where each laminate is represented by its averaged eigenstrain and compliance as is elaborated in section 4.1 and section 4.3 respectively.

- The crystal lattice is oriented randomly in each grain. This means that the anisotropic elastic constants obtained from ab initio calculations are assigned to each individual coordinate system.

- Surface and interface energies are accounted for (semi-)analytically.

Given the above conditions it is not possible to determine a priori which transformations result in the lowest specific transformation barrier $e_{b_{min}}$ that in turn minimizes the specific free energy Δg of the whole system (=RVE). The main problem is that the contribution of the elastic strain energy $u_{vi} = \frac{U_{vi}}{V_i}$ to e_b plays a major role and it is not clear how to (if possible at all) reliably predict u_{vi} and consequently that specific grain-laminate pair which minimizes the next energy increment. This is outlined in section 5.1. Obviously a strain energy minimum can be found by calculating all combinations of micro structural possibilities for a certain fraction of transformed grains and then choosing the minimum from all these results. However, such an approach would (i) not be efficiently calculable because of the immense number of necessary jobs (calculable by means of combinatorics) and (ii) it would not produce a realistic dynamic evolution of the product phase. Therefore an algorithm must be developed which efficiently and realistically minimizes the free energy upon transformation while keeping the total number of calculations low enough. An incremental approach is chosen, where only on grain is subject to transformation within the increment. However, in this increment U_{vi} is explicitly calculated for all not-transformed grains and their laminates using the finite element method. Then the grain-laminate pair with the lowest $e_{b_{min}}$ minimizing the systems Δg is picked out and represents the energy minimizing state of this increment. Note that the increments correspond to the transformation of one grain respectively but should not be associated with a time as is elaborated in section 5.3. This procedure is repeated until all grains of the RVE have transformed, see Figure 3.6.

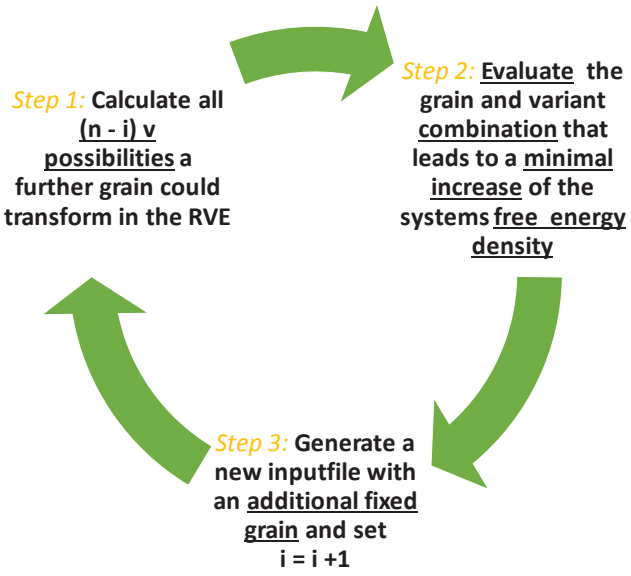

Figure 3.6.: Illustration of a transformation increment. In a transformation increment one energetically favorable martensite grain is chosen to transform. This is repeated until a full martensitic state is reached. n...total number of grains in the RVE, i...number of already transformed grains, v...number of martensite laminates

Denoting the total number of grains in the RVE n, the number of already transformed grains i and the number of martensite laminates v the total number of calculations carried out in the above procedure is $\sum_{i=0}^{n}(n-i)v \simeq (n+1)\frac{n}{2}v$. However, this number can significantly be lowered by preselecting grain variant combinations that are known to be more likely to transform from previous calculations because they are not too far away from the current transformation site. In the algorithm here presented there are two parameters describing the degree of preselection. First, the fraction of all possible states representing the grain-laminate combinations that are more likely to transform and second the number of increments that are carried out using that preselected set. Based on the results from a full analysis where all $(n-i)v$ possibilities have been calculated in each increment these parameters can be estimated. Using the estimated preselection parameters the number of necessary calculations is one order of magnitude less then without using any such strategy, while it has been verified that the results are identical.

3.6. Preprocessing and Mesh Optimization

In this section the mesh preprocessing of the random and regular microstructures are discussed as the mesh plays a very important role considering computation time as well as validity of the calculated strain energies. Also the creation of some text files which are fixed parts of the input-files as well as the input-file's general structure and dynamically rewritten parts are discussed. At the end, specifications of simulation parameters in the code are given.

Since the whole model is defined on a nanoscale all lengths are considered in nanometers. Note that Abaqus has no built-in system of units. Therefore all input data must be specified in consistent units. In section B.1 the system of consistent units on the nanoscale considered here and some relations to other useful units are given. In both the regular and the random microstructure model tetrahedral elements with quadratic shape functions are used (C3D10) in a nonuniform mesh with elements of slightly different size. However, much effort is spent on generating a nearly uniform mesh. Tests indicated that for a proper selection of grain-laminat pairs it is sufficient to run the algorithm described in subsection 3.5.2 with a relatively coarse mesh. This is crucial because the time for the high number of calculations necessary in the algorithm to find a realistic minimum of the free energy can be significantly reduced since the computation time scales quadratically with the number of elements. However, to obtain proper values for the strain energy the determined sequence of grain-laminate pairs is recalculated with a very fine mesh ensuring a valid result for the strain energy.

First, the creation of the regular RVE is elaborated. It is completely accomplished using CAE. For the construction of a truncated octahedron it is emphasized that it is the Voronoi diagram of the b.c.c cubic lattice.After meshing, the necessary node-sets for the periodic boundary conditions according to Figure A.2 must be selected manually. Then, in order for the periodic boundary conditions to be written for the node-sets, the sets must me ordered so that matching nodes are at the same position in the nodeset, which is achieved using another python script that sorts each node-sets for the coordinate it has the largest dimension in.

As previously mentioned, for the creation of the random micro structure the software "Neper" is used. The exact execution parameters for the coarse and for the fine mesh respectively can be found in Appendix D. The output from the polycrystal generator is modified partially manually and partially via a python script, which uses the information of a generated statistics file to rename the generated element sets to matrix sets and inner connected grain sets constituting the evaluated grain-cluster RVE. After renaming the matrix sets are easily combined and rewritten using CAE.

In both cases the results of the mesh preprocessing, which are the nodes, elements and sets of either are written together with a set of randomly generated orientations into a file.

This file is the first part of every input-file during a full simulation and therefore termed the "static" part of the input-file. Other static input-file sections saved in separate text-files are the material data and job specific data or in the case of the truncated octahedra RVE the periodic boundary equations. All these can be found in Appendix C. Figure 3.7 provides an overview of static and dynamic input-file sections. Finally, the parameters for the simulation, i.e. the names of the static files as well as the boundary conditions have to be specified at the beginning of the main script found in section E.1.

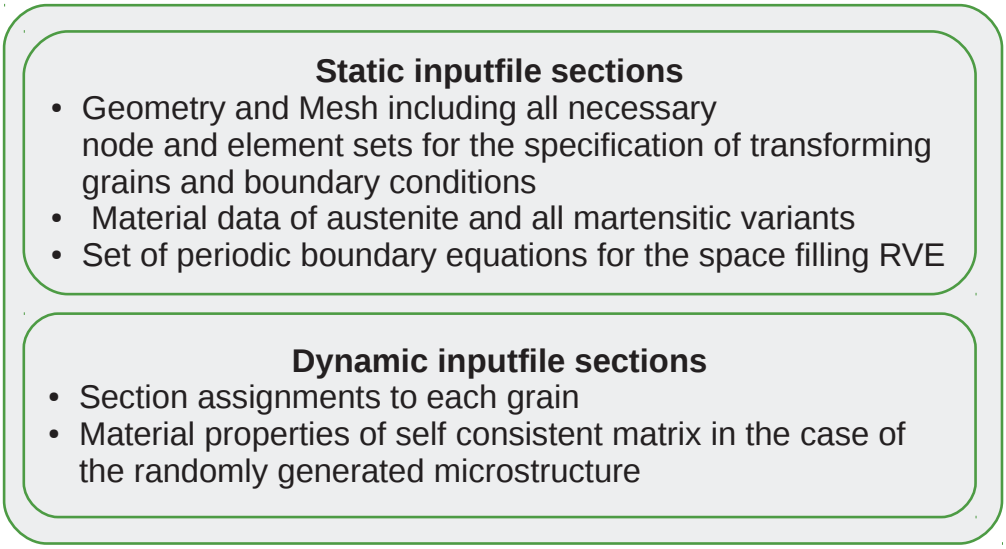

Figure 3.7.: Overview of static and dynamic input-file sections. Each static section is saved in a seperate textfile, that does not change during the whole simulation. Parts of the input-file are dynamically rewritten in each increment of the simulation based on previous results.

3.7. Postprocessing

In this work the postprocessing of the simulation is limited to the evaluation of the generated text-files. For the main simulation these are: save_grain, save_model, allruns_n where n is the increment number. The first two hold only the information of the energy minimizing state determined in each increment for the evaluated grain-laminate pair and for the whole model, respectively. The file allruns_n holds the information of all calculations carried out in the increment n. save_grain holds the volumes and specific energies that are strain energies as well as IEs for the energy minimizing grain-laminate pair, whereas save_model holds cumulative data of volumes and energies, i.e. the fraction of austenite and martensite respectively as well as the sum of energies in the whole RVE. Note that always volumes and either a specific or total energy corresponding to that volume is saved so that that third component can be calculated easily for weighting purposes.

4. Material Data

4.1. Transformation Strains

In this section the strain energy calculation of transformation strains used as input for the finite element calculation of U_v is discussed. Note that in the model here presented the transformation strain is applied as anisotropic thermal expansion in the finite element program. Experimentally it is observed that a single laminate structure of two martensite variants is the dominating morphology in nanograined NiTi with grain sizes under $\approx$ 100nm. The experimentally observed laminate structures consist of certain compound twins, with an average variant fraction of $\mu = 0.5$ and an average variant width of $d \approx$ 1.5nm. Therefore, not all theoretically possible laminates consisting of two twin variants $\mathbf{U_i}, \mathbf{U_j}$ according to subsection 2.5.1 are considered in this model, but only those observed experimentally. Note that $\mu \approx 0.5$ is only observed for a purely thermal transformation. This no longer hold if an external load provides additional driving force. For the multi-variant model to be efficiently calculable the transformation strains of the martensite variants forming a laminate were averaged resulting in the transformation strain of the entire laminate. It is hence not necessary to resolve single variants in the FE discretisation. The lost information of the effects of the opposing shear strains is taken care of by Σ_{TW}. In Table 2.1 the deformation gradients $\mathbf{F_i}, \mathbf{F_{i'}}$ forming a laminate can be found. Interestingly, the six observed laminates consist of all 12 unique deformation gradients possible in NiTi. Since for NiTi in a purely thermal transformation $\mu = 0.5$ the average or mean deformation $\mathbf{F_m}$ then is:

$$\mathbf{F_m} = 0.5(\mathbf{F_i} + \mathbf{F_{i'}}) \tag{4.1}$$

The general Green - St-Venant or Lagrangian finite strain tensor $\mathbf{E}$ is used, because in the non linear theory large rotations can be accounted for and the twins modeled here are rational (hence generic).

$$\mathbf{E} = \begin{pmatrix} \epsilon_{11} & \epsilon_{12} & \epsilon_{13} \\ \epsilon_{21} & \epsilon_{22} & \epsilon_{23} \\ \epsilon_{31} & \epsilon_{32} & \epsilon_{33} \end{pmatrix} = \frac{\mathbf{1}}{\mathbf{2}} \left(\mathbf{F^T F} - \mathbf{I} \right) \tag{4.2}$$

However, Abaqus interprets expansion strains as enginerring strains. It was verified that the shear components of $\mathbf{E}$ are only slightly different from those calculated by the linearized small strain theory. Therefore, as a first approximation the shear terms ϵ_{ij} are set equal to those in the small strain theory where the tensorial shear strain components of the strain tensor can then be expressed using the linearized strain definition: $\epsilon_i = \frac{\partial u_i}{\partial x_i}$ and $\gamma_{ij} = \gamma_{ji} = \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}$. The results of the averaged laminate strains are shown in Table 4.1.

Table 4.1.: Symmetric deformation tensors (transformation strains) of the martensite variants. Note that Abaqus uses the engineering strain definition for anisotropic expansion.

$\underline{\underline{\epsilon}}_t^i$	$\underline{\underline{\epsilon}}_t^1$	$\underline{\underline{\epsilon}}_t^2$	$\underline{\underline{\epsilon}}_t^3$	$\underline{\underline{\epsilon}}_t^4$	$\underline{\underline{\epsilon}}_t^5$	$\underline{\underline{\epsilon}}_t^6$
ϵ_{11}	0.0234	0.0234	0.0234	0.0234	-0.0381	-0.0381
ϵ_{22}	-0.0381	0.0234	-0.0381	0.0234	0.0234	0.0234
ϵ_{33}	0.0234	-0.0381	0.0234	-0.0381	0.0234	0.0234
$\gamma_{12} = 2\,\epsilon_{12}$	0	0.1186	0	-0.1186	0	0
$\gamma_{13} = 2\,\epsilon_{13}$	-0.1186	0	0.1186	0	0	0
$\gamma_{23} = 2\,\epsilon_{23}$	0	0	0	0	-0.1186	0.1866

4.2. Interface Energies

In this section the values used for the IE model described in section 3.1 are given. A value of $F_c = 5.8 \cdot 10^6\ J/m^3 = 5.8 \cdot 10^{-12}\ nJ/nm^3$ of the irreversible work of friction was taken from the measurements of the thermodynamics of the transformation of coarse-grained NiTi [61]. However, it is known that F_c has to be much higher in nanograined NiTi, consequently the above value is only a lower bound. Experimentally an irreversible energy of 3 J/g was measured for an estimated maximum fraction of 64% martensite using differential scanning calorimetry (DSC) [35]. Considering the density of NiTi being 6.45 g/cm^3 and assuming the work of friction as the only irreversible dissipated energy contribution yields an F_c of $(1.935 \cdot 10^7)/0.64\ J/m^3 \approx 3 \cdot 10^{-11}\ nJ/nm^3$ which would be about 5 times higher than in coarse grained NiTi. Note that F_c does not change the transformation kinetics but only the onset of the transformation, thus the results can be compared afterwards for both values as is discussed in section 5.5. For Σ_S a slight increase of 0.1 $J/m^2 = 0.1 \cdot 10^{-9}\ nJ/nm^3$ was estimated, since it is supposed that the transformation causes misfit at the grain boundary, thus $\Sigma_m > \Sigma_a$. Σ_S can only be provided with high uncertainty and must therefore be treated as open parameter in the model. However, Σ_S must not be greater than $0.4 \cdot 10^{-9}\ nJ/nm^2$. A specific twin-boundary energy Σ_{TW} of 0.014 $J/m^2 = 0.014 \cdot 10^{-9}\ nJ/nm^2$, was calculated by first-principle calculations of atomic models of compound twins, based on HRTEM experiments [62].

4.3. Elastic Constants

Previous investigations of the stored elastic strain energy in NiTi nano structures using isotropic material data and a single grain morphology [41] underestimated the experimentally measured elastic strain energy for two reasons. First, the isotropic Young's modulus of martensite obtained from tensile experiments differs quite strongly from NiTi B19' real elastic constants due to detwinning effects influencing the measurement as is explained in subsection 1.1.2. Second, the effect of accommodation (also explained in subsection 1.1.2), which is considered to be of great importance is not captured using a single grain model. As a remedy, a set of fully anisotropic elastic material constants (inverse compliance tensor) for the B19' phase is computed by DFT (Density functional theory) [45] in an ordered NiTi phase and given in Voigt notation (4.3). An overview of the full elastic tensor of all 81 entries corresponding to 13 independent constants in the case of a monoclinic system (21 in the most general case of a triclinic system) can be found in appendix E.4. The calculated constants were compared to two other sets recently published by Wagner and Windl [63] as well as Hatcher [64], where the elastic constants of the different phases were calculated using pseudo-potential and FLAPW (full potential linearized augmented planewave). Possible shortcomings of these works are the lack of lattice relaxation [45]. Interestingly in both works softening of the elastic constants during the forward martensitic transformation is a result of the ab initio studies [63, 64]. Insignificant differences regarding the constant's absolute values are observed. The ab initio elastic constants used here for B19' are given in the coordinate system of the tetragonal cell as is shown in Figure 2.4 a). In general, crystal properties like elastic properties are expressed by a matrix, depending on the choice of both the crystal axes and the Cartesian reference frame. Therefore, comparison of calculated elastic constants with results of other calculations or experimental data is only possible provided the chosen crystal axes and reference frame are identical. In [45] the orientation of the monoclinic cell of martensite was chosen such that the unique axis c lies along Z and γ is associated with the angle between +a and +b (see also subsection 2.4.1). The elastic constants matrix for any monoclinic phase then has the general form:

$$\begin{bmatrix} \bullet & \bullet & \bullet & 0 & 0 & \bullet \\ & \bullet & \bullet & 0 & 0 & \bullet \\ & & \bullet & 0 & 0 & \bullet \\ & & & \bullet & \bullet & 0 \\ & & & & \bullet & 0 \\ & & & & & \bullet \end{bmatrix}$$

where • are non zero entries. This general form for the martensite was calculated to be:

$$\underline{\underline{C}}^m = \begin{pmatrix} 254 & 104 & 136 & 0 & 0 & 21 \\ & 180 & 151 & 0 & 0 & 0 \\ & & 248 & 0 & 0 & -6 \\ & & & 91 & -3 & 0 \\ & & & & 93 & 0 \\ & & & & & 5 \end{pmatrix} GPa \tag{4.3}$$

Below, the general convention of elastic constants in Voigts notation used in Abaqus is shown. It is mentioned that it is a little different to what is commonly found in literature, where the index pairs 23 and 12 are interchanged.

$$C_{ijkl} = \begin{pmatrix} C_{1111} & C_{1122} & C_{1133} & C_{1112} & C_{1113} & C_{1123} \\ C_{2211} & C_{2222} & C_{2233} & C_{2212} & C_{2213} & C_{2223} \\ C_{3311} & C_{3322} & C_{3333} & C_{3312} & C_{3313} & C_{3323} \\ C_{1211} & C_{1222} & C_{1233} & C_{1212} & C_{1213} & C_{1223} \\ C_{1311} & C_{1322} & C_{1333} & C_{1312} & C_{1313} & C_{1323} \\ C_{2311} & C_{2322} & C_{2333} & C_{2312} & C_{2313} & C_{2323} \end{pmatrix} GPa$$

The fact that the entry $C_{2223} = C_{26}$ in $\underline{\underline{C}}^m$ is zero is not due to some symmetry but is only a result of the ab initio calculation. For the austenite phase it is not reasonable to use ab initio values yet, since most of the calculations are still carried out at 0K. At this temperature the austenite phase would be unstable, causing the calculation to yield unreasonable results. Moreover, B2 is a cubic phase with very weak anisotropy because of the high cubic symmetry, which makes isotropic data a very close reasonable approximation. Additionally, for temperatures notably above M_s stress induced martensite, which would otherwise lead to a misinterpretation of the Young's modulus, is no longer expected. The isotropic Youngs modulus was determined as 65 GPa and a Poisson's ratio of 0.4 was used in agreement with works by Fukuhara et al. [65] as well as Mei et al. [66]. Finally the austenite was assumed to be stress free prior to transformation.

To compare the anisotropic elastic constants for B19' to isotropic ones from literature the script written for the micro-mechanical self-consistence scheme was adopted to a higher number of random orientations (10^6) as compared to the RVE investigated here (100-200) in order to improve the averaging of the anisotropic compliances as is explained in section 3.3. In order to obtain isotropic reference values (Youngs modulus (E) and Poissons ratio (ν)) for the martensite the so obtained elastic tensor C^m_{ave} given below is compared to the isotropic representation of Hooke's law, given in Equation 4.4. A system of two equations obtained by equal entries of the general and averaged elasticity tensors

must be solved for E and ν.

$$\underline{\underline{C}}^m_{ave} = \begin{pmatrix} 255 & 120 & 112 & 0 & 0 & 7 \\ & 254 & 136 & 0 & 0 & 0 \\ & & 220 & 0 & 0 & 0 \\ & & & 60 & -4 & 0 \\ & & & & 60 & 0 \\ & & & & & 45 \end{pmatrix} GPa$$

$$\begin{pmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \tau_{xy} \\ \tau_{xz} \\ \tau_{yz} \end{pmatrix} = \frac{E}{(1+\nu)(1-2\nu)} \begin{pmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{pmatrix} \begin{pmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \gamma_{xy} \\ \gamma_{xz} \\ \gamma_{yz} \end{pmatrix} \quad (4.4)$$

Following this procedure the elastic constants are determined as $E^m = 110$ GPa and $\nu^m = 0.33$ for B19' martensite. Common values of Youngs modulus from literature are in the range of 55-80 GPa for austenite and 25-45 GPa for martensite. Obviously, the data calculated above for martensite suggests that martensite is actually the harder phase. It even differs by a factor of 3-4 from common literature values. However, it is pointed out that DFT calculations yield ground-state properties of materials at 0K, such that temperature effects are not taken into account, hence elastic constants are usually overestimated. However, it is assumed that the tendency of martensite being harder than austenite is correct and that the effect of detwinning in the martensite phase described in subsection 1.1.2 is responsible for the much lower measured values obtained from literature [63].

For the simulation the elastic tensor $\underline{\underline{C}}^m$ in (4.3) was averaged the same way as the transformation strains. Therefore, it was first rotated into the cubic austenite basis by a clockwise rotation around Y in Figure 2.4 a). Then $\underline{\underline{C}}^m$ is referred to the same coordinate system as $\mathbf{F_a}$ or $\mathbf{U_a}$ (subsection 2.4.3). Next, the rotations in the cubic symmetry group relating $\mathbf{F_a}$ and $\mathbf{F_i}, \mathbf{F'_i}$, respectively, are determined. Note that since there are two rotations $\mathbf{R}$ in the cubic symmetry group relating $\mathbf{F_a}$ to $\mathbf{F_i}$ there are four combinations of rotations that describe the laminate $\mathbf{F_i}, \mathbf{F'_i}$. It was verified that all four so obtained averaged elastic tensors are equal within a tolerance < 10 GPa. All averaged elastic laminate tensors can be found in appendix C.1.

5. Results and Discussion

In this chapter the obtained results are discussed and compared to other models as well as experimental data. First, the IEMTA is compared to other transformation strategies. Second, the energy evolution is discussed at the grain scale, which must be considered for the transformation criterion and the RVE scale from which averaged macroscopic quantities can be derived. Then the importance of the strain energy contribution to the free energy is emphasized and discussed. Next, a transformation kinetics is derived from the results of the simulation, followed by a discussion on the impact on varying model parameters such as some IE parameters. Finally, the amount of the experimentally observed retained austenite is discussed. It is pointed out that in this chapter the results of the regular microstructure RVE with periodic boundary conditions (PBC) and the random microstructure RVE embedded in a self consistent matrix ($ESCM$) are presented and compared to each other.

5.1. Comparison of Transformation Strategies

In this section the effectiveness of the incremental energy minimization is demonstrated. To this end, two alternative transformation strategies are introduced: (i) A random transformation is considered, where "random" means that for a certain number of grains, which are determined to transform, the grain and the laminate are chosen arbitrarily. In further consequence this transformation strategy is termed *random toggling*. (ii) Prior to the IEMTA, as a first approach to avoid the still high number of calculations necessary in the IEMTA, an incremental transformation strategy, based on a local transformation criterion (LTC) has been considered. Here in each increment the transforming grain-laminate pair is determined from the current state of the RVE in the following way: First, each grain is assigned an averaged stress tensor $\langle \underline{\underline{\sigma}}^i \rangle$ (average over all its integration points see section E.3). This effective stress tensor is considered in combination with the 6 calculated transformation strains $\underline{\underline{\epsilon}}_t^i$ to define the transformation driving force $\langle \underline{\underline{\sigma}}^i \rangle : \underline{\underline{\epsilon}}_t^i$. Considering all IEs, the grain-laminate pair fulfilling $max[\langle \underline{\underline{\sigma}}^i \rangle : \underline{\underline{\epsilon}}_t^i - (\Gamma_{IN} + \Gamma_{TW} + \Sigma_S)\frac{A_i}{V_i}]$ is chosen to transform in each increment. The evolution of the strain-energy density following this transformation criterion compared to random toggling is shown in Figure Figure 5.1. For a better illustration of this transformation criterion, the curve of the RVE's strain energy density (black) is additionally split up into the martensite (blue) and austenite (red) phase respectively. A comparison of the black line with results from a random toggling (black points) shows that the transformation criterion based on the mechanical driving force is not sufficient to predict a transformation evolution that accommodates the evolving strain energy best, because it lacks important information from outside the grain domain considered in the LTC, e.g. which phases are the nearest neighbors of a certain grain or if the neighboring grains are orientated advantageously, such that they are deformed more easily. Looking, at the strain energy evolution in each phase, it can be seen that at lower martensite fraction the strain energy is much higher in the martensite phase and much lower in the austenite phase compared to a random transformation. Obviously, the formation pattern of the martensite using this transformation criterion is an agglomeration of the evolving martensite phase, where the martensite forms a growing cluster of grains. If the IEMTA is used, however, there is no agglomeration of the martensite phase, but a network-like growth, where in the early stages of the transformation the martensite forms a net of connected martensite variants around yet untransformed austenite regions as can be seen in Figure 5.2. Note, that this effect is better illustrated in the *PBC* model because of the limited RVE sizes. In any case, the explicit calculation of the arising strain energies is considered necessary after this comparison. Next, the results of the strain energy minimization using the IEMTA are presented. As can be seen in Figure 5.4 the IEMTA reduces the arising strain energy significantly compared to the random toggling.

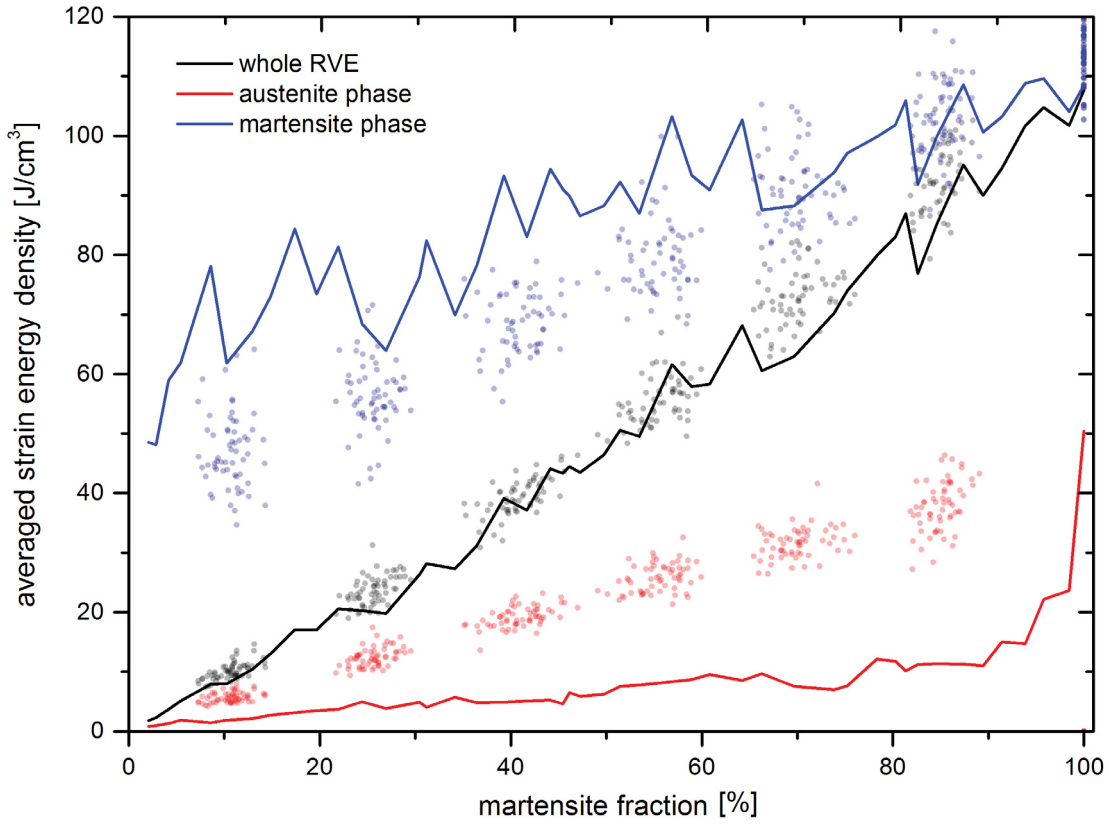

Figure 5.1.: Comparison of the strain-energy evolution for the LTC (lines) and random toggling (points). The first leads to a strong agglomeration of transforming grains, hence the strain energy increases faster in the martensite phase than in the random toggling. From the black line it can be seen that with this transformation criterion the strain energy is not better accommodated than by means of a random toggling.

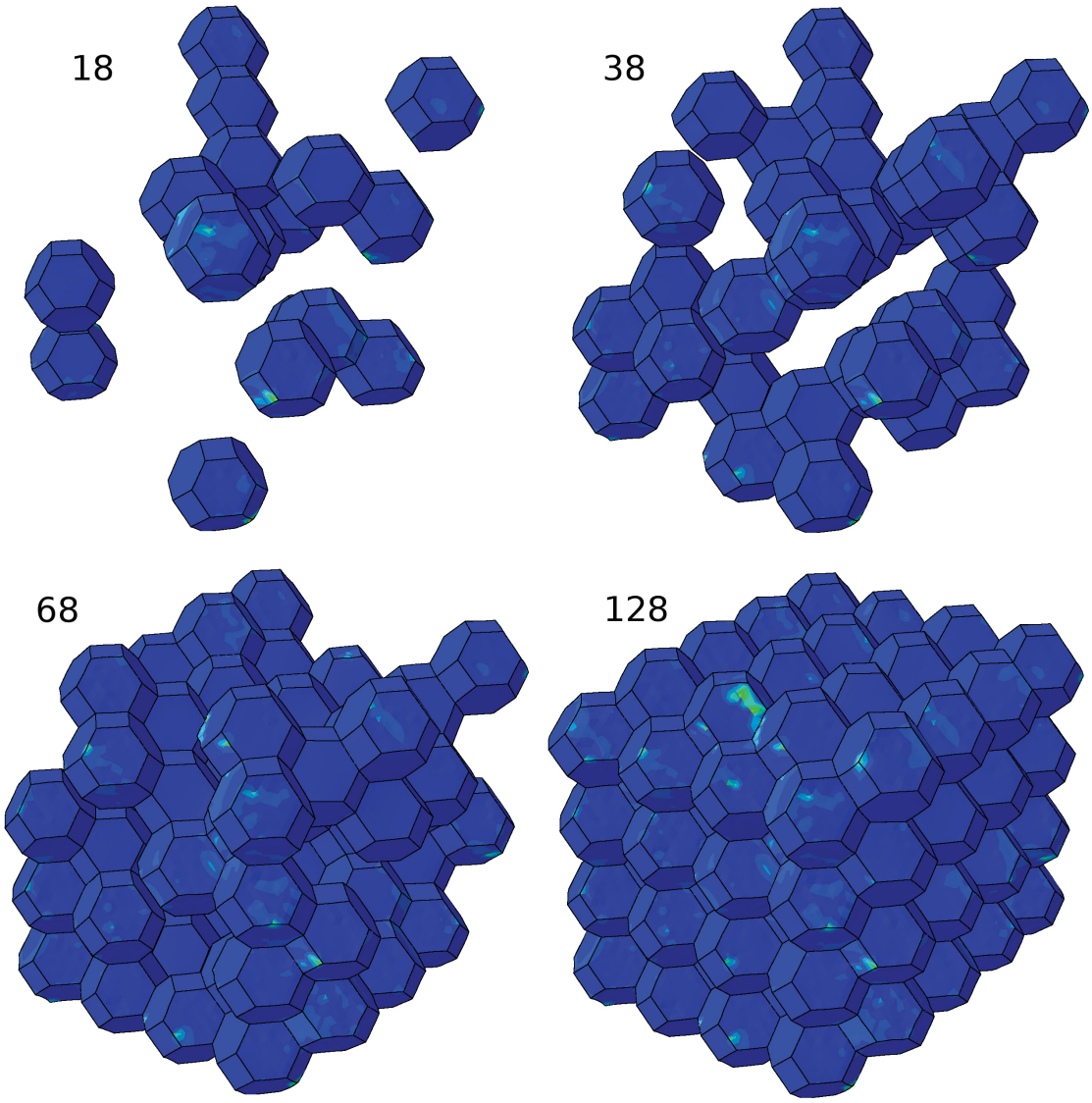

Figure 5.2.: Illustration of the evolution of martensite in the *PBC* model, consisting of 128 grains. The numbers indicate how many grains have transformed. From the upper states it can be seen that the martensite forms a network of connected grains which has already spread over the entire RVE for a martensite fraction of about 30%. Note that grains at the border are connected by the PBCs.

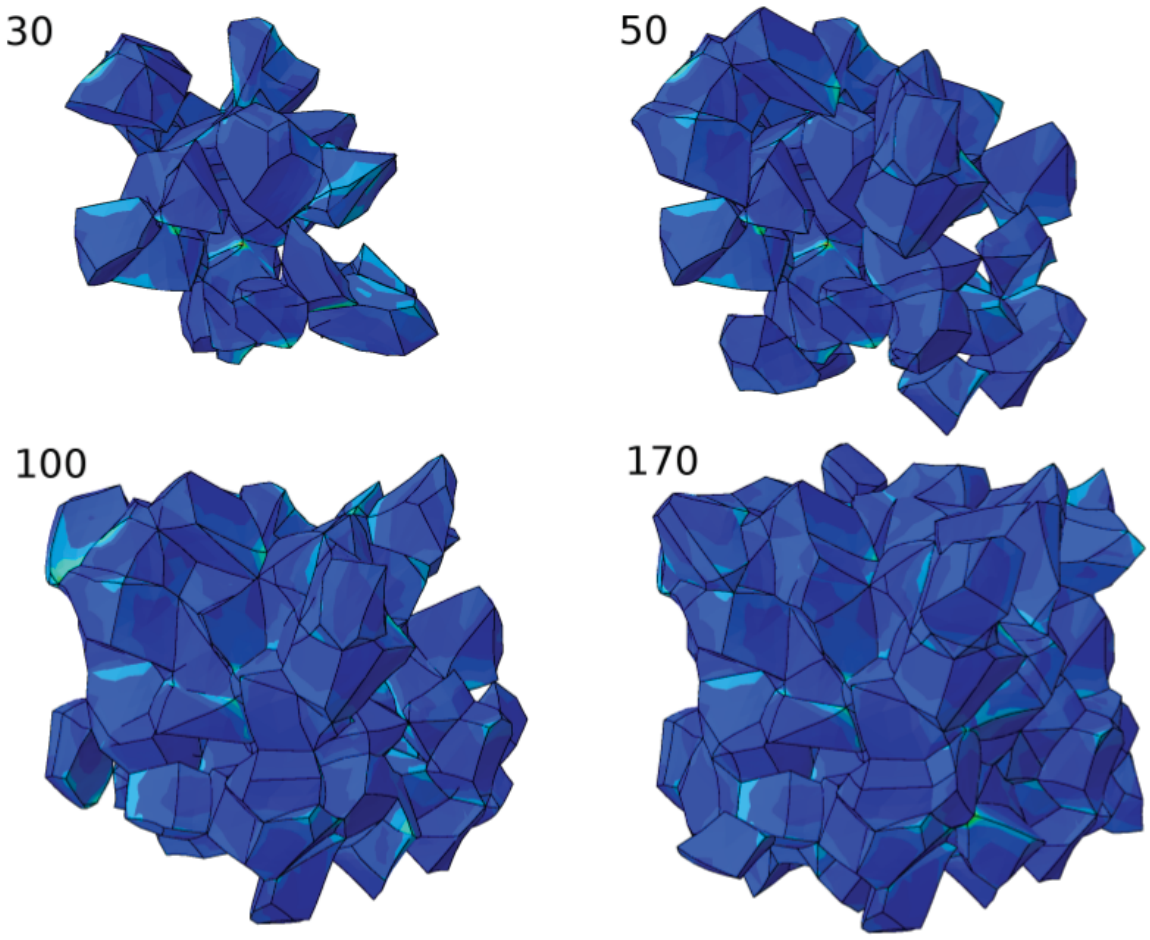

Figure 5.3.: Illustration of the evolution of martensite in the *ESCM* model, consisting of 170 grains. The numbers indicate how many grains have transformed. From the upper states it can be seen that a network-growth is present but less pronounced than in the *PBC* model since the self-consistent matrix does not transfer stresses from outside the RVE.

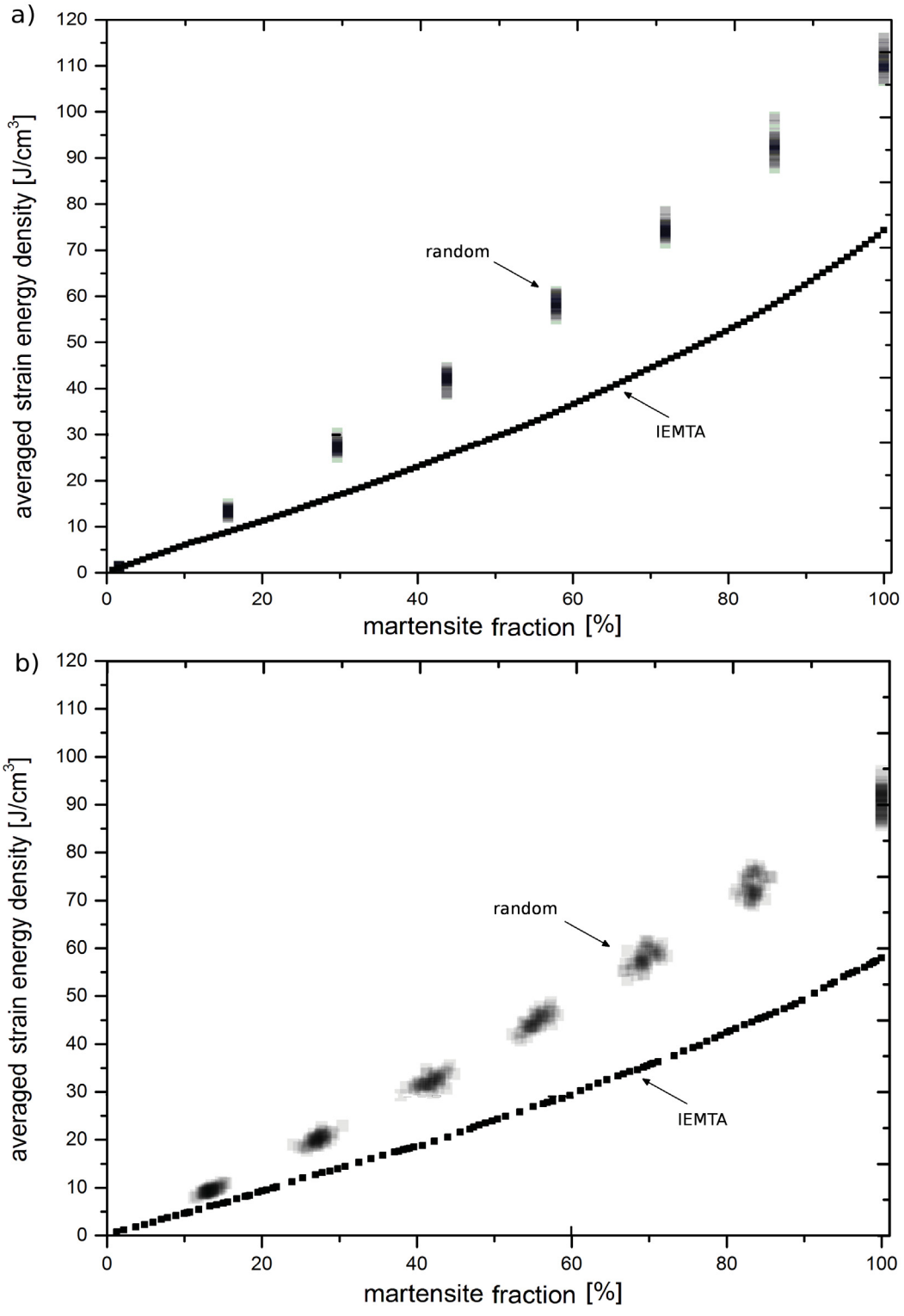

Figure 5.4.: Comparison of the strain energy evolution for the IEMTA and the random toggling. a) *PBC* model b) *ESCM* model. In both models, using the IEMTA, the produced strain energy density is reduced by one third compared to the random toggling.

5.2. Energy Evolution

First, the calculated energies are locally investigated at the grain level, since at this level the transformation criterion is defined. In Figure 5.5 the calculated minimal energy barriers e^i_{bmin} for each transforming increment and their compositions are visualized for the *ESCM*. It can be seen that the strain energy constitutes the main portion of the transformation barrier, even larger than the sum of all other contributions. Considering the high influence of the strain energy, obviously the martensitic transformation is facilitated at free surfaces, because there strain energy is lower. Furthermore, the contribution of the strain energy to the energy barrier increases on average as the transformation proceeds. This means that the strain energy is mainly responsible for the transformation rate to decrease as the transformation proceeds contrary to the common assumption in the literature, that the transformation is suppressed for smaller grains mainly because of their higher IE. This discussion is continued in section 5.6 where the input parameters for this model are discussed. Another noteworthy fact is that the transformation barrier is not necessarily monotonously increasing with progressing transformation. This will be important in the subsequent two sections.

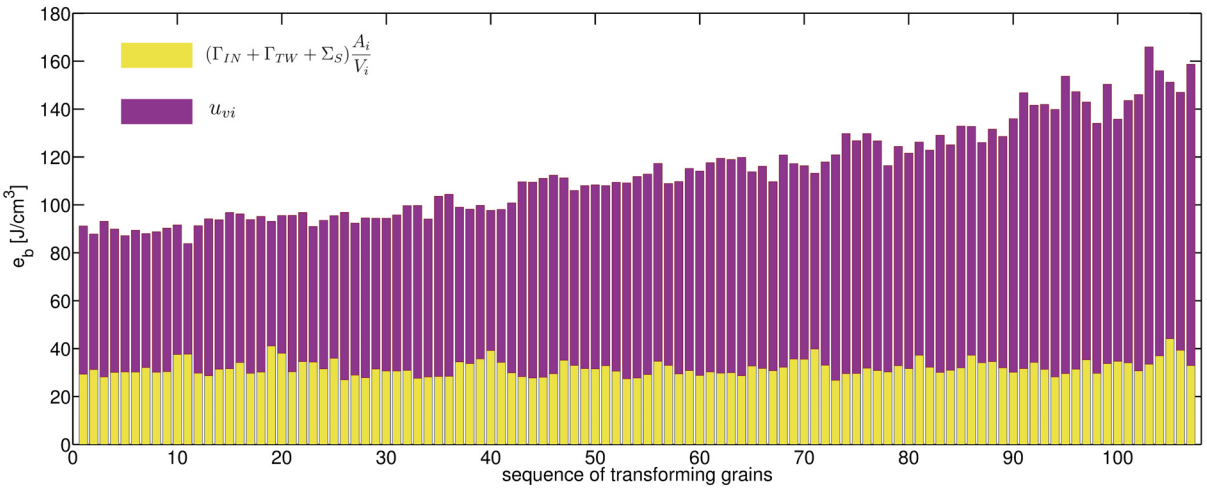

Figure 5.5.: Illustration of the contributions to e^i_{bmin} for the determined energy minimizing sequence of grain-laminate pairs in the *ESCM*. The yellow bars indicate the contribution of all IEs and the stacked purple bars the contribution of the strain energy. It can be seen that (i) the strain energy contribution is dominating and (ii) grows on average in the course of the transformation.

Now the calculated energies are investigated at the RVE level. A new variable e_m is introduced representing the sum of all arising IEs and strain energies due to the martensite transformations relative to the total RVE volume. In Figure 5.6 this variable is plotted versus the martensite fraction. Experimentally, a value of $\approx 5\ J/g$ reversible strain energy is measured using DSC at an estimated maximum of 64% martensite [35]. Considering the density of NiTi (6,45 g/cm^3) this yields a value of about 32 J/cm^3. Then it has to be considered that the IE term Σ_{TW} actually represents the shear related part of the strain energy. Σ_{TW} accounts for about 35% of all IEs with the parameters used here (see section 4.2). From Figure 5.4 a) it can be seen that for 64% martensite phase the strain energy density of the *PBC* RVE is 40 J/cm^3, while e_m is 58 J/cm^3 according to Figure 5.6. The difference of 20 J/cm^3 is obviously attributable to IE terms. However, 65% of these 20 J/cm^3 i.e. $0.65 \cdot 20 = 13\ J/cm^3$ are non-mechanical IEs, which must be deducted in order to make it comparable to the experiment. Consequently, for a martensite fraction of 64% the found strain energy for the *PBC* model is $58 - 13 = 45\ J/cm^3$. For the random microstructure likewise an additional IE of about 13 J/cm^3 has to be considered, yielding a strain energy of $50 - 13 = 37\ J/cm^3$, which comes close to the experimentally measured value. The fact that both results are greater than the experimentally measured value indicates that the elastic constants from ab inito are overestimated. Assuming that the behavior of a realistic microstructure can be found between the two models the strain energy and consequently the elastic constants are probably overestimated by 20%.

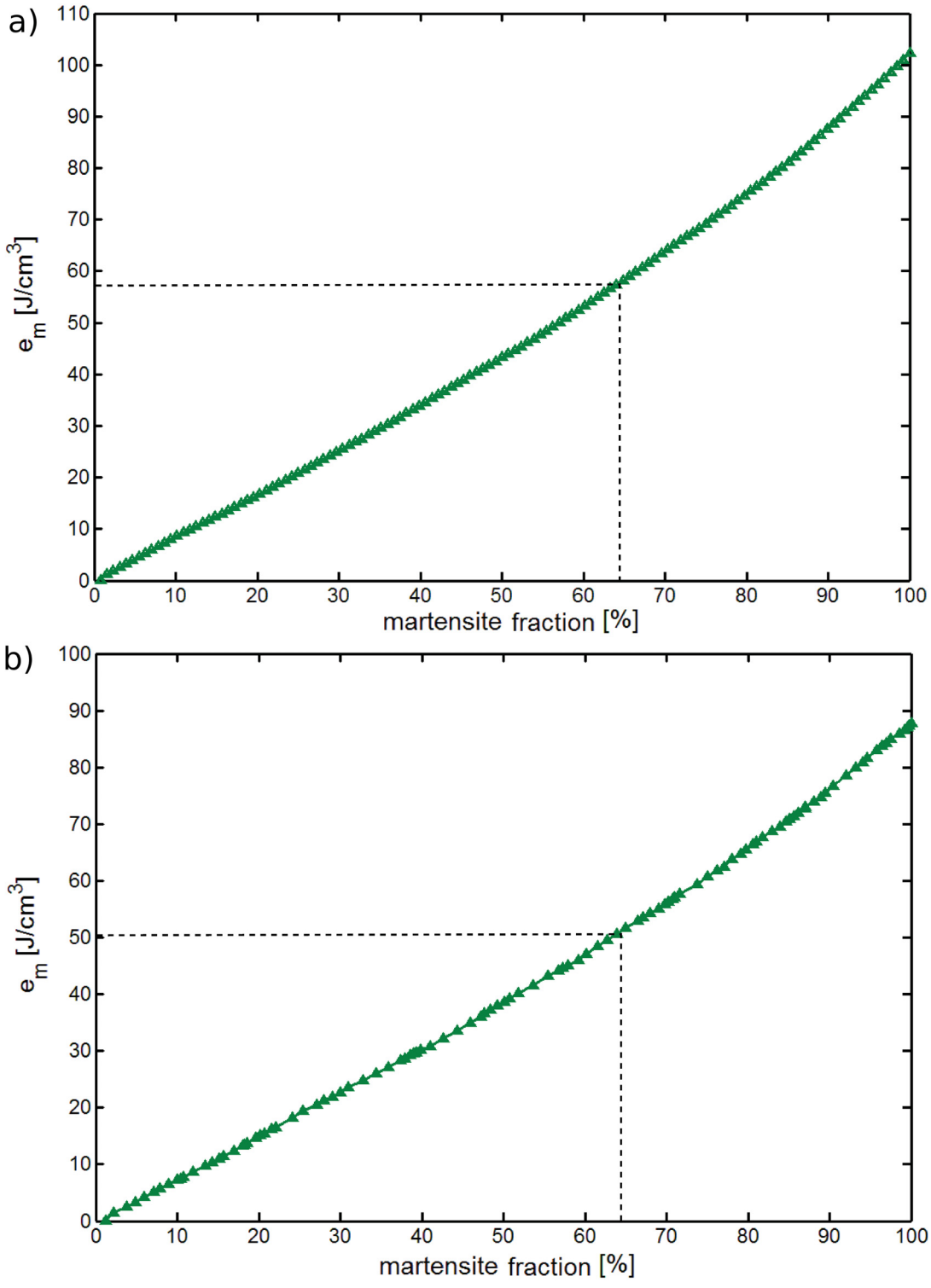

Figure 5.6.: Illustration of the total energies e_b versus martensite fraction evolving in the course of the transformation.a) PBC model. b) ESCM model. The experimentally measured value of reversible strain energy [35] is near the value in the *ESCM* model.

5.3. Stress Influence on the Transformation

The initial minimum energy barrier in Figure 5.5 is about 90 J/cm^3. If the specific driving force has reached that level, i.e. $-\Delta g = e_{bmin}$, the first transformation occurs. Since it is well known that the velocity of a martensitic transformation is only limited by the speed of sound, it occurs much faster than any change of $\Delta g(T)$ necessary to cause another thermally induced transformation elsewhere in the material. The following case is observed: Some transformations cause an advantageous stress field, meaning that the induced stress field partially annihilates existing stress fields of previous transformations, consequently reducing the produced strain energy u_{vi}. In some cases this even causes $e^{i+1}_{bmin} < e^{i}_{bmin}$ as pointed out in the previous section. Therefore in the next increment this grain automatically transforms without any further increase of Δg. It has to be pointed out that in an increment of the IEMTA there could be several grains for which this holds true. The transformation is said to be self-triggering. Note that since self-triggering is a consequence of an energy minimization, it also reflects the ability of self accommodation. So far a martensitic transformation proceeds under isothermal conditions. However, since the energy barrier increases on average, at some point it is again necessary to further increase Δg (i.e. to apply further cooling in the case of the forward transformation) in order to trigger further transformation. Note that the necessary Δg to start the first transformation ($\approx$ 90 J/cm^3) is larger than the interval of the transformation itself ($\approx$ 80 J/cm^3). The results of the simulation are shown in Figure 5.7 and Figure 5.8 for each model.

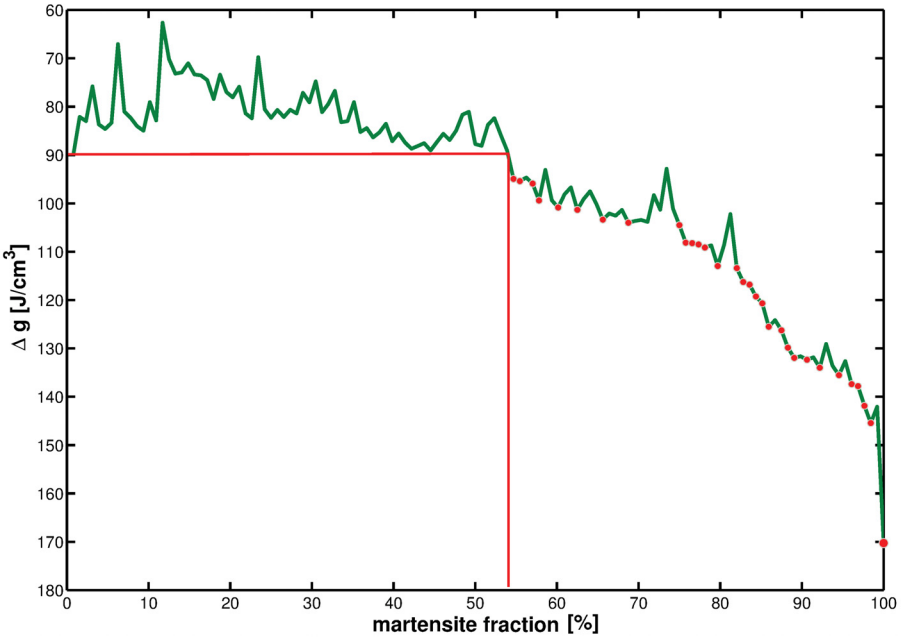

Figure 5.7.: Illustration of self-triggering for the *PBC* model. After the transformation has started the arising stress field causes an autocatalytic transformation until there is more martensite than austenite. The red points indicate the levels of chemical driving force that have to be reached to cause further transformation.

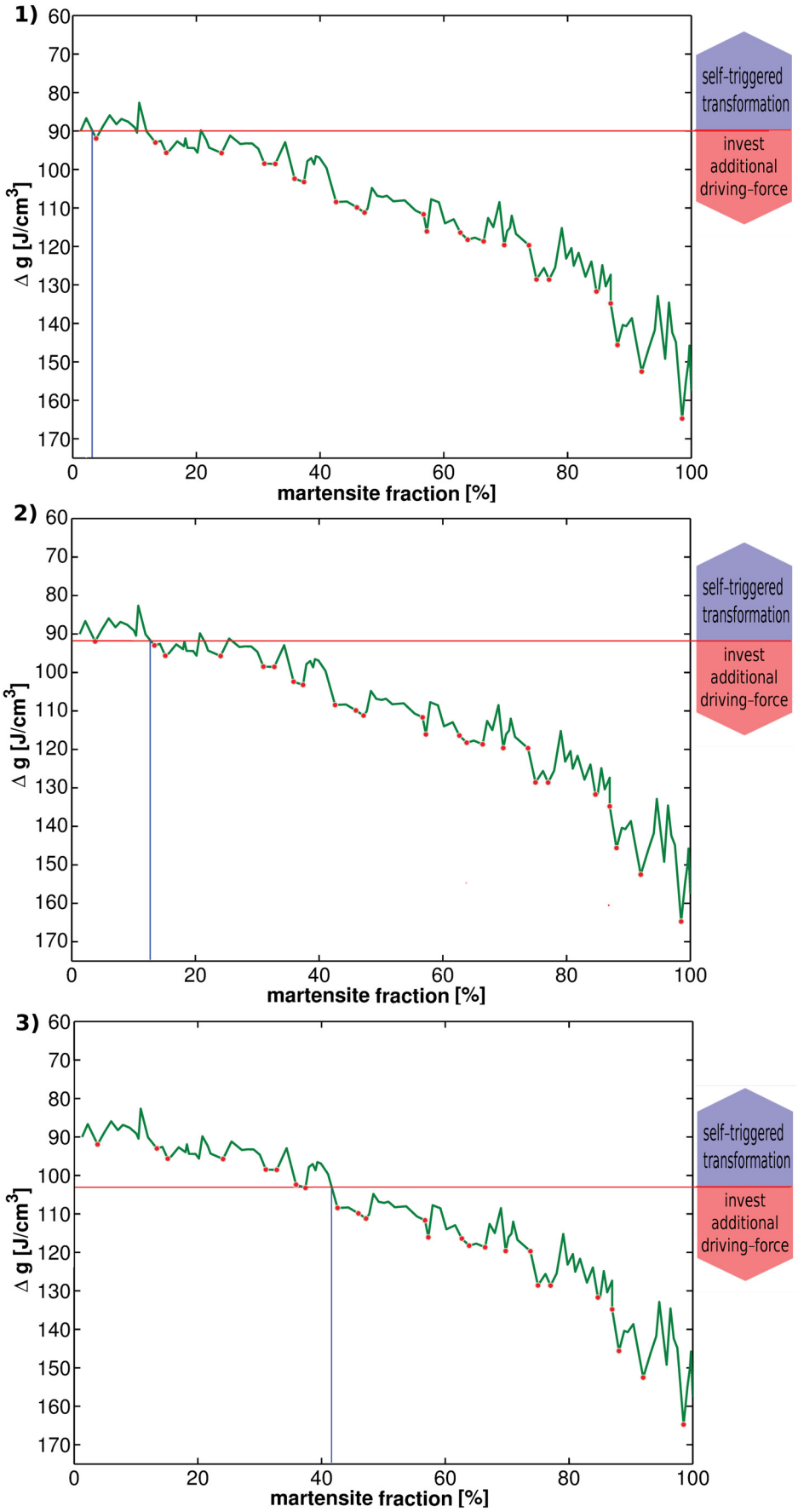

Figure 5.8.: Visualization of the process of the minimum free energy barrier for the found grain using the IEMTA in the *ESCM* model. Exemplary levels of chemical driving force and the according martensite fraction are indicated.

5.4. Transformation Kinetics

According to Figure 5.8 every increment in driving force Δg creates a certain number of new martensitic grains thus increasing the overall martensite fraction. Plotting this fraction versus the invested driving force Δg yields the transformation kinetics from an energetic point of view as illustrated in Figure 5.9. For the *PBC* model the self-triggering effect in the initial state is more pronounced, because the boundary condition enables transformation triggering in all directions starting from the initial nucleation site, whereas for the *ESCM* model the triggering effect is only obtained in one direction for grains next to the matrix. The self-triggering effect in the *PBC* model proceeds until a phase fraction of about 50% martensite has formed, which again reflects the network growth of the martensite discussed in section 5.1. For both models the highest increase of martensite can be found at the same level of the chemical driving force, because of the same average grain diameter of the equivalent sphere in both models. Δg for the initial transformation in the *ESCM* model is 5 J/cm^3 less than in the *PBC* model because of the more complex morphology and because there are occasionally larger grains than in the *PBC* causing a lower IE barrier. In both models the required Δg to complete the martensitic transformation until no retained austenite is left is approximately equal. As already discussed in section 3.4 at this small RVE scale a realistic macroscopic behavior is found somewhere in between both models. Note that for much larger RVEs the behavior of both models must converge to a uniform behavior because of the much higher volume to surface ratio. As a next step the obtained kinetics is compared to common kinetics models. Kinetics of martensitic transformations are often reproduced by phenomenological approaches for a fast implementation in numerical algorithms. For athermal martensitic transformations probably the most popular example is the well known Koistinen Marburger kinetics [67]. It was initially obtained by fitting X-Ray diffraction measurements of volume-% fraction retained austenite as a function of the quenching temperature interval in ferrous martensite. The exponential fit function is of the form: $\xi = 1 - e^{-C(M_s - T_q)}$, where C is a constant, M_s is the already discussed martensite start temperature and T_q is the current quenching temperature. A minor shortcoming of the Koistinen Marburger kinetics is that it lacks a small incubation period that is observed in real transformations. Assuming a linear relation between the temperature and the chemical driving force in the vicinity of the transformation, the obtained kinetics is fitted using the same exponential ansatz: $\xi = 1 - e^{-C(\Delta g^1 - \Delta g)}$. Note that since the time for a martensitic transformation is much less than the time the chemical driving force changes enough to cause a transformation elsewhere in the RVE, only the maximum values of the martensite phase for a certain driving force are considered. The constant C for the *PBC* and *ESCM* model is $C_{\mathrm{pbc}} = -0.0387$ and $C_{\mathrm{escm}} = -0.563$, as well as $\Delta g^1_{\mathrm{pbc}} = 90\ J/cm^3$ and $\Delta g^1_{\mathrm{escm}} = 89\ J/cm^3$ respectively.

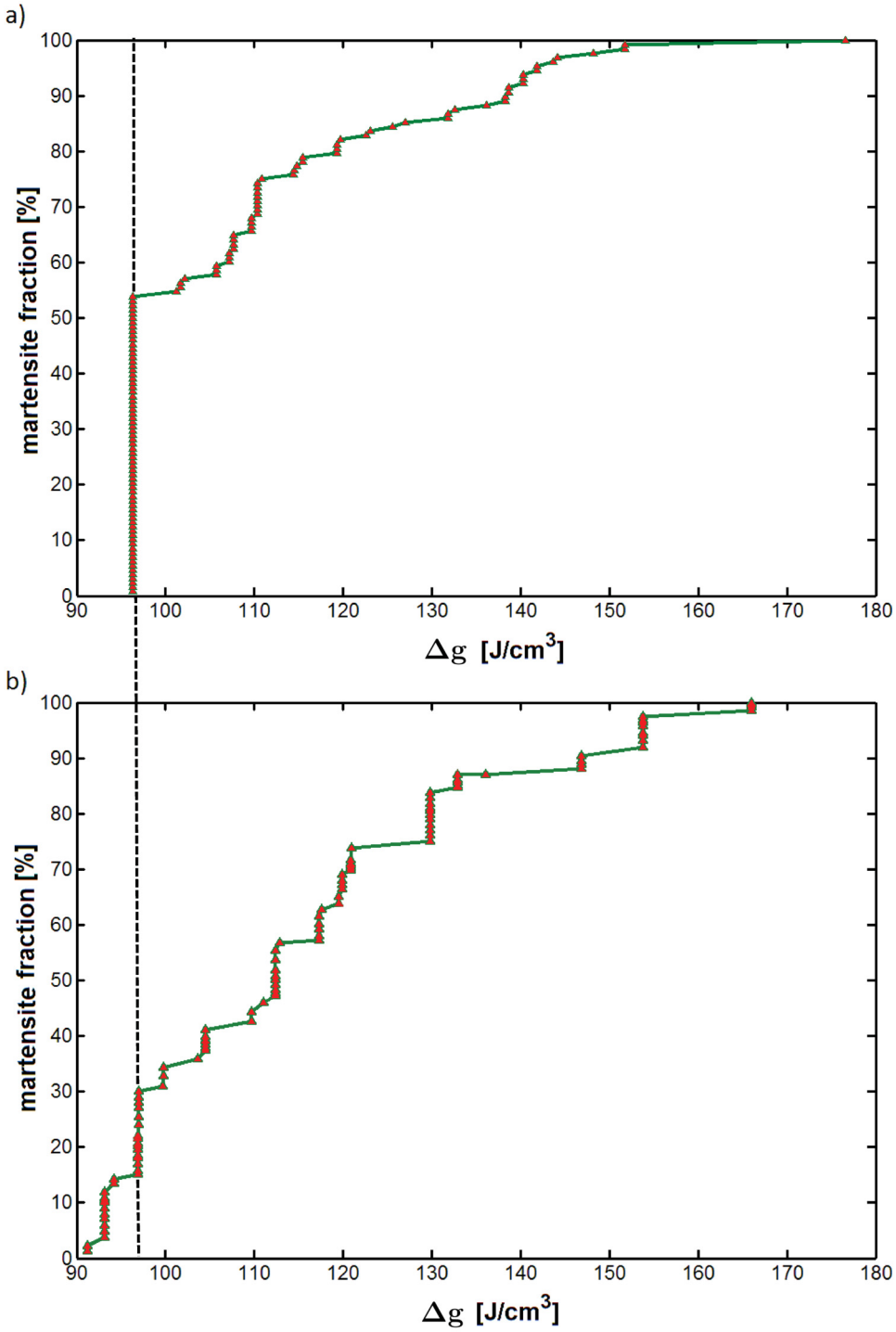

Figure 5.9.: Kinetics of the transformation in terms of energies. a) *PBC* model. b) *ESCM* model. The self-triggering effect is most pronounced in the initial transformation of the *PBC* model. The minimum energy barrier in the *ESCM* model is lower than in the *PBC* model. Interestingly, the highest increase of martensite fraction occurs at the same level of chemical driving force.

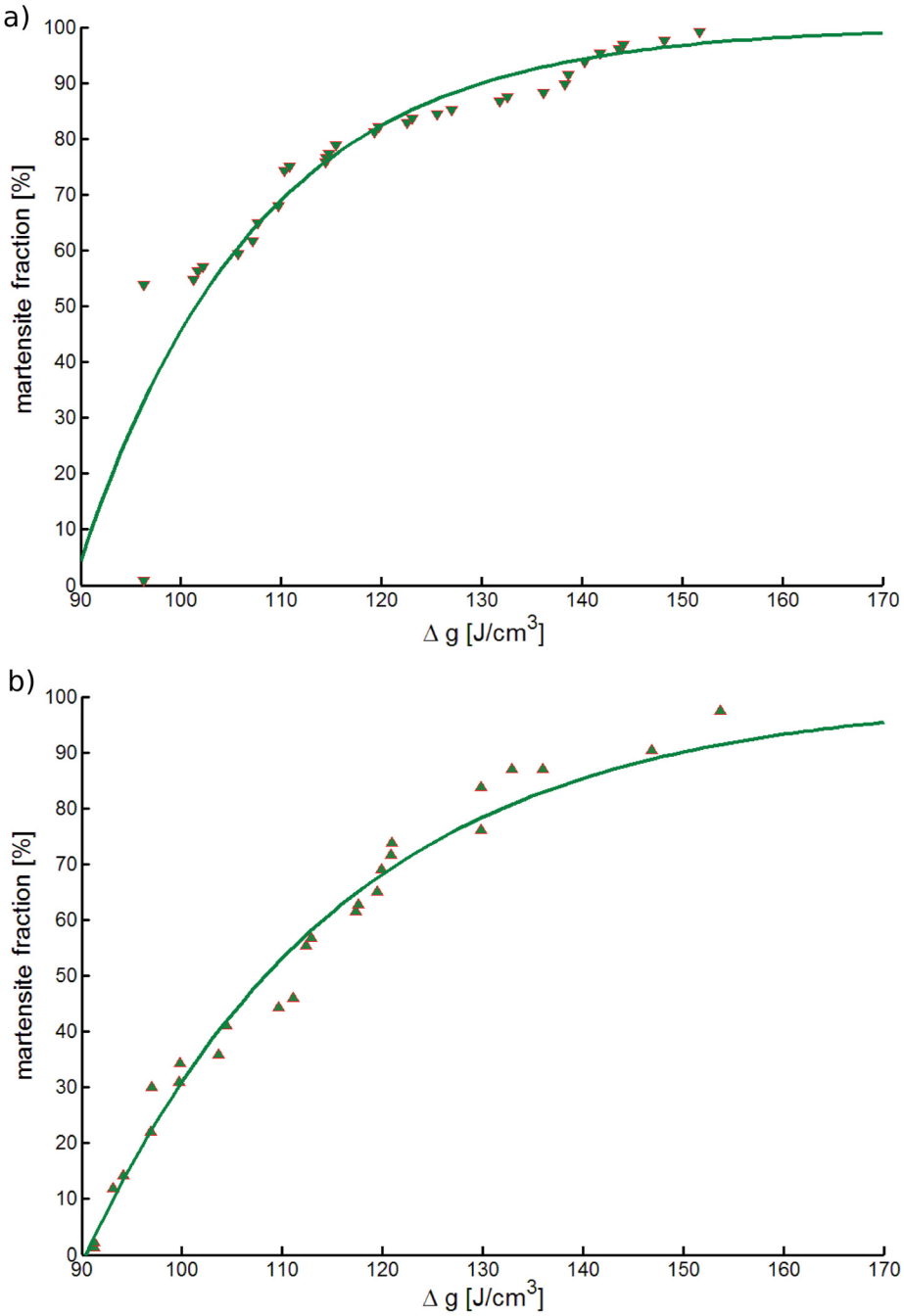

Figure 5.10.: Koistinen Marburger fits of the obtained transformation kinetics. a) *PBC* model. b) *ESCM* model.

5.5. Open Parameters of the Energy Barrier

As mentioned in chapter 4 the parameter Σ_s can only be estimated with a high degree of uncertainty. In the above simulations Σ_s is assumed to be 0.1 J/m^2. Because this parameter gives rise to a scaling with δ^2 it is responsible for the suppression of the transformation in smaller grains. For the average grain size considered here Σ_s constitutes about 20% of the energy barrier without u_{vi}. The highest reasonable value of 0.4 J/cm^3 results in a 25 J/cm^3 larger energy barrier for the average grain size. A reference simulation is run with $\Sigma_s = 0.4$. As can be seen from Figure 5.11 the higher IE does not only delay the transformation by the discussed 25 J/cm^3, but also changes the course of the transformation. However, the effect is not pronounced because of the generally low contribution of Σ_s to the total energy barrier. Note that for the regular tessellation the curve would only be shifted without any change because of the equal grain sizes. Another critical point is that the work of friction F_c is massively underestimated. Note that F_c does not change the course of the transformation since it is assumed to be equal for all grains. However, assuming that the value for F_c for coarse-grained NiTi is indeed five times smaller than that for nanograined NiTi, the scaling behavior of F_c should also be investigated. Furthermore, in the theory for the determination of Σ_{TW} so far only isotropic material properties have been assumed. However, NiTi obviously has a considerable anisotropy, which should not be neglected.

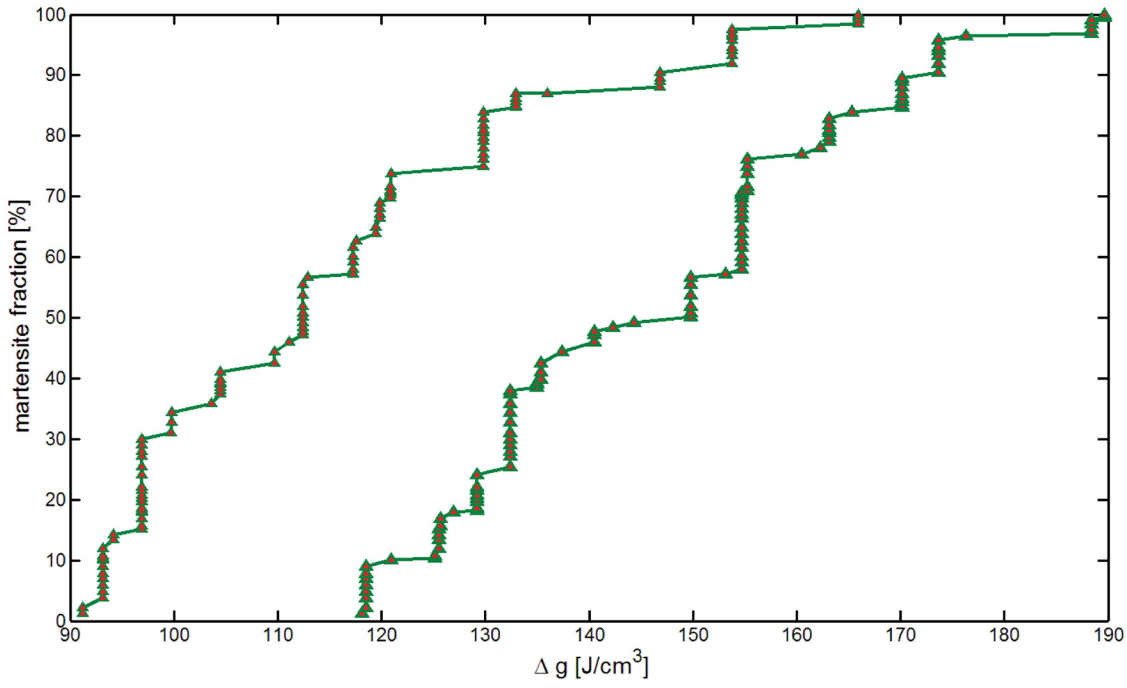

Figure 5.11.: Influence of a higher value of Σ_s on the transformation kinetics for the *ESCM* model. A higher value of Σ_s shifts the transformation to higher driving forces and causes a different transformation path.

5.6. Retained Austenite

In this section the experimentally observed retained phase beside martensite is discussed. It is consists of austenite and R-phase, but for the sake of simplicity in the following discussion it is referred to as retained austenite. First, the facts following [35] are summarized: (i) Contrary, to the transformation in coarse-grained NiTi in nanocrystalline NiTi the transformation is strongly suppressed and occurs via an intermediate R-phase. (ii) A quantitative determination of the fraction of retained austenite is hardly possible. A maximum fraction of 64% transformed martensite is estimated from energy measurements using DSC [35]. (iii) An estimation of the transformation temperature for grains of 80 nm diameter yields about -75 K. Thermodynamically, the equilibrium temperature T_0 between austenite and martensite can be approximated by $\frac{1}{2}(M_s - A_f)$ [68] yielding a value of 89°C, measured by DSC in nanograined NiTi. According to the literature the difference of entropy between the two phases is 0.5 J/cm^3. A linear extrapolation of $\Delta g = (T_0 - T)\Delta s$ is only valid in the vicinity of the equilibrium temperature. However, in the case of NiTi the required undercooling is so large that a linear extrapolation becomes irrelevant. Nevertheless, since $s \longrightarrow 0$ for $T \longrightarrow 0$ it can be seen as an upper boundary. Therefore, from the initial specific energy barrier $e_b^1 = 90 J/cm^3$ in the simulation, the start of the transformation would occur at $89 - 90/0.5 = -91$°C. Obviously, the calculated energy barrier is too high. Furthermore, it must be mentioned that the much higher work of friction is not even considered in this calculation. This again indicates that the calculated elastic constants are overestimated. Therefore, the calculation of elastic constants above 0 K is crucial.

6. Conclusions and Outlook

The effect of self-accommodation has been quantified in nanograined NiTi using an incremental energy minimizing transformation algorithm. The automation and optimization of the IEMTA enables to study RVE sizes, which already capture important characteristics of the martensitic transformation on higher scales. As has been discussed in section 5.3 on the one hand the strain energy is found to be the dominant contribution to the energy barrier, on the other hand it can trigger the transformation at a constant temperature. Furthermore, the simulation indicates that the transformation barrier mainly grows because of the evolving stress state. Therefore the strain energy mainly determines the course of the transformation. However, as explained in section 3.1 in this model only a lower boundary of the IE has been considered. The IE model used here increases significantly for very small grains [54]. Consequently, the course of the transformation is more affected by the IEs if smaller grains are present. Therefore, clearly a broader distribution of grain sizes must be considered for the artificially generated RVE. The fact that the measured reversible energy is slightly lower than the results of the simulation indicates that the elastic constants from ab inito are overestimated. Considering the observed network-growth of the martensite phase, it is reasonable that experimentally a maximum fraction near the estimated value of 64% martensite is obtained, since at this fraction the amount of additional driving force for the transformation increases stronger, because the remaining austenite is surrounded by the harder martensite. A direct correlation of the obtained energies to temperatures is not possible, because the function of $\Delta g(T)$ is not available in this large temperature interval.

Possible extensions to the model presented in this work along with a collection of ideas for some alternative applications are enumerated as follows:

- Triggering of the transformation by applying external forces to the RVE in the vicinity of the free energy equilibrium to study the differences of thermal as well as stress-induced transformation. It would be interesting to test how large a uniaxial loading has to be, so that a transformation is energetically favorable.

- Use of preferred orientation data e.g. from pole figures of textured NiTi as in [69]. Due to the literature texture has a strong influence on the selection of the martensite

morphology and hence on the SME [70].

- Adjustment of the grain morphology to stretched or plate-like grains as they are obtained by rolling for instance.

- Extension to a non-homogeneous temperature field where the chemical driving force becomes a function of the position, representing for instance contact with a hot material.

- Extension to a non stress free initial state. Real microstructures are not exactly stress-free since phase boundary motion always requires a certain driving traction. The presented model can easily be extended to a non stress-free initial state in the first step.

No reverse transformations are included in the transformation model even though in certain cases they may be energetically favorable. However, due to the shift between $M_s - A_f \simeq M_f - A_s$ that is attributable to the work of friction the chance for this to happen is considered low.
Another noteworthy aspect to be taken into account in future research is the existence of the intermediate R-Phase. Considering this phase would lead to other energy minimizing transformation paths. However, the deviation from the evolution found here is assumed to be small because of the small produced strain energy upon transformation to this phase.
The model can be used to calibrate IE contributions like the energy of the austenite martensite IE. Also the study of transformation fatigue may be a prospective application by evaluating a full transformation and subsequently introducing artificial defects in the microstructure, which are then considered in a recalculation.
The exact same model could be employed to investigate the reverse transformation caused by a heating process, i.e. the transformation from martensite to austenite.
In a future work the investigation of the effect of elastic anisotropy on the twin scale has to be investigated. The effect of coupling terms between shear stresses and normal strains as well as normal stresses and shear strains must be investigated. Finally, the question how the twin laminate forms in an arbitrary grain morphology remains to be solved.

A. Periodic Boundary Conditions

Periodic boundary conditions are an efficient way to obtain mean quantities of relatively large systems by modeling comparatively small ones. As previously mentioned, the requirement for PBCs to be applicable is that the modeled region is space filling. This applies for the RVE consisting of truncated octahedra shown in Figure 3.4. In a periodic tesselation for each of the RVE's surfaces there is at least one corresponding face on the other side(s). These faces are equally sized in a space-filling pattern, separating two adjacent RVEs. Note that for PBCs to be applicable in a finite element model also the RVE's mesh has to be periodic. This means that the nodes of the surface's mesh have to be congruent. Each node of the node-set has to be coupled with its matching node of the corresponding node-set. In the finite element software Abaqus [60] this is accomplished with the *EQUATION command in the input file, where the overall deformation behavior of the RVE is governed by virtual reference points. The necessary equations for PBCs can be derived from

$$\begin{pmatrix} \epsilon_{11} & \epsilon_{12} & \epsilon_{13} \\ \epsilon_{21} & \epsilon_{22} & \epsilon_{23} \\ \epsilon_{31} & \epsilon_{32} & \epsilon_{33} \end{pmatrix} \begin{pmatrix} \Delta l_1 \\ \Delta l_2 \\ \Delta l_3 \end{pmatrix} = \begin{pmatrix} \Delta u_1 \\ \Delta u_2 \\ \Delta u_3 \end{pmatrix} \tag{A.1}$$

where of ϵ_{ij} represents the overall average engineering strains, the vector containing Δl_i is the metric distance of two coupled node-sets and the vector containing Δu_i is the difference of the displacements of the coupled node-sets. The equation coupling the node-sets "faceFront" and "faceBack" is explained exemplarily. To facilitate the following procedure the RVE is rotated such that as many Δl_i as possible become 0. This is the case if the RVE's faces are parallel to the global coordinate system as shown in Figure A.3. Then, since $\Delta l_2 = \Delta l_3 = 0$ the first equation obtained from Equation A.1 is simply $\epsilon_{11}\Delta l_1 = \Delta u_1$, where $\Delta u_1 = u_1(faceFront) - u_2(faceBack)$. For the implementation of the equation in the inputfile, it has to be rewritten as:

$$u_1(faceFront) - u_1(faceBack) - \epsilon_{11}\Delta l_1 = 0 \tag{A.2}$$

Furthermore, it is common practice to introduce virtual reference nodes to hold the information of the overall deformation of the RVE necessary to define the equations. In this work they are referred to as REFD and REFS, where the degrees of freedoms (DOFs) of

REFD correspond to normal strains and the DOFs of REFS to shear strains. Equation A.2 is added to the inputfile as:

$$faceFront, 1, 1., faceBack, 1, -1., REFD, 1, -1., \tag{A.3}$$

Until now only the coupling of surfaces has been described. However, in the case of a space-filling tesselation this must also be done for lines and corners that bound these surfaces. Corresponding lines and corners that have to be coupled appear at least three times on each space-filling RVE. To get a clearer picture of the couplings it is helpful to imagine a cubic tessellation that fills the space. Opposite faces of the cube have to be coupled. Each edge of the cube has to be coupled with three other edges and each corner of the cube must be coupled with each of the other ones. Figure A.2 gives an overview of all couplings necessary to obtain full PBCs. Figure A.3 shows projections of the RVE to comprehend the couplings. The coloring in both figures refers to the given legend. The set of equations presented here can be used for every cell consisting of $2n^3$ grains, where n is an arbitrary number. Note that a periodic mesh could in principle also be achieved for a randomly generated microstructure in the following way: (i) Fixing all lines and corners of the periodic tesselation. (ii) Generating a random surface, bounded by the fixed lines and corners, for one side of the RVE. (iii) Translating the information of the surface mesh created in (ii) to the opposite side of the RVE and use it as a constraint for the generation of the mesh at this side. This way a fully periodic mesh can be generated. However, at the time of this thesis, such an algorithm was not available. In Figure A.1 the effect of the implemented PBCs is illustrated.

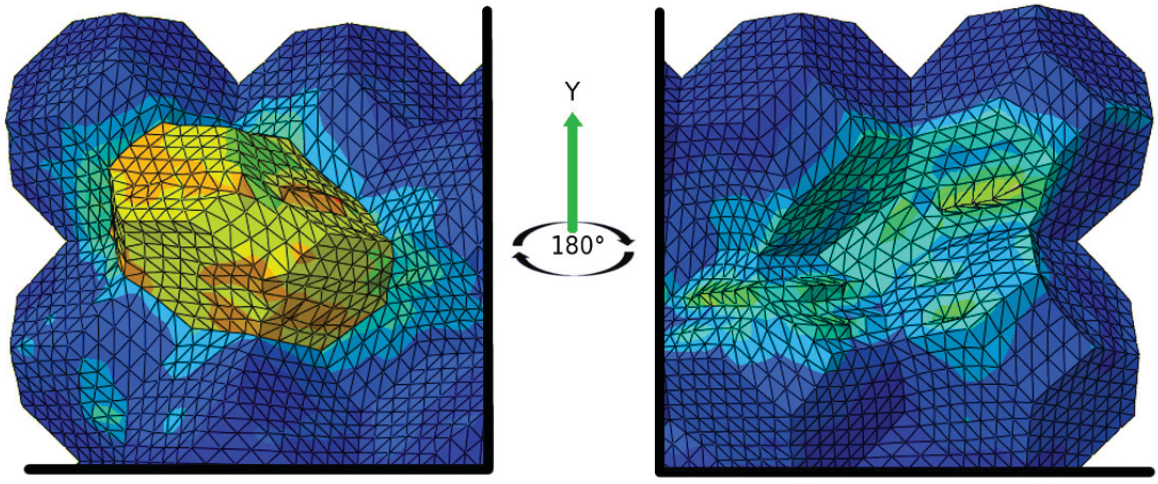

Figure A.1.: Illustration of the periodic bounary conditions. The grain on the left hand side is transformed and creates a certain deformation. At the other side of the RVE this deformation is perfectly reproduced complementary so that the RVE remains space filling

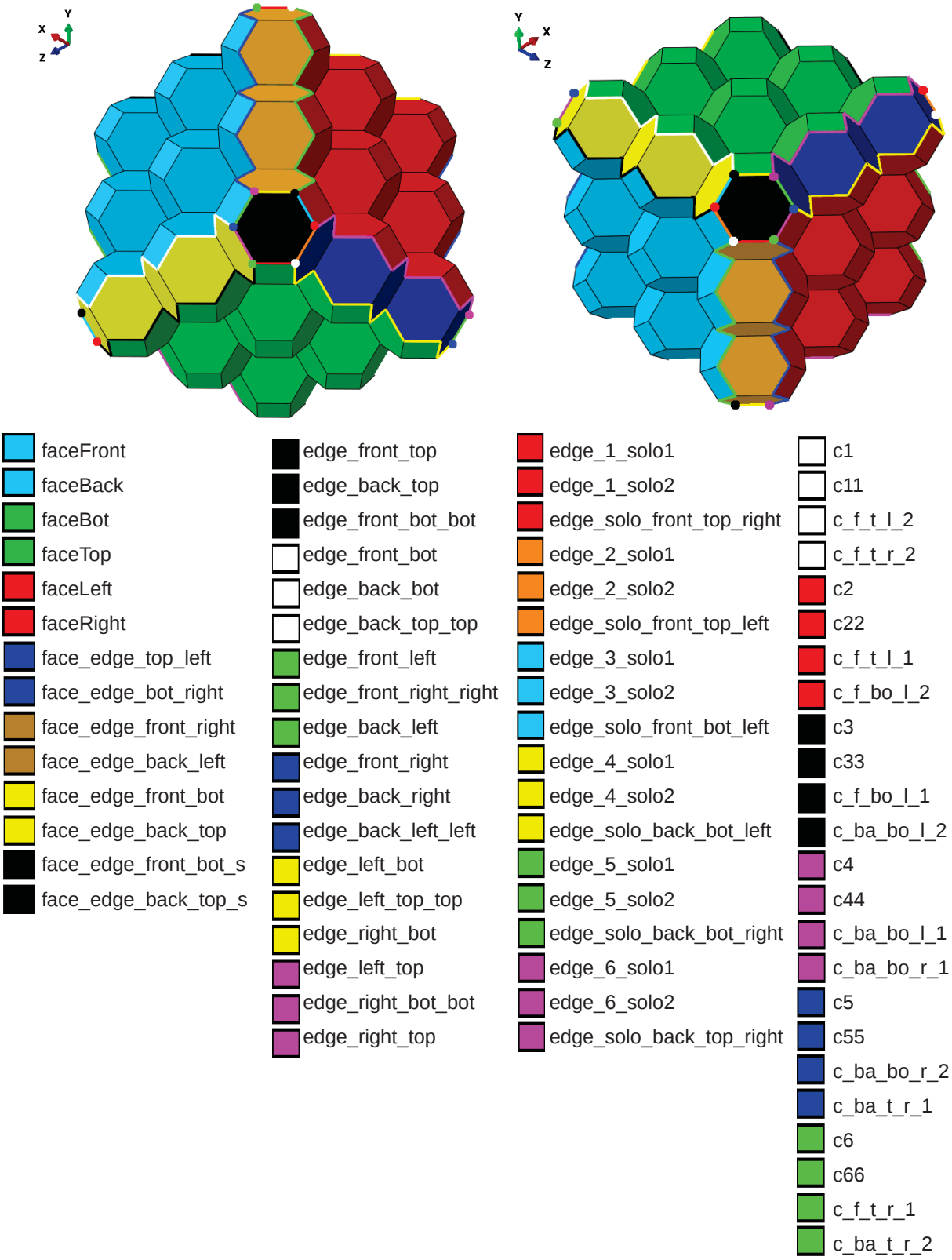

Figure A.2.: oblique projections of the modeled cell and a colored legend of all used node-sets for the setup of the periodic boundary conditions (Also see Figure A.3).

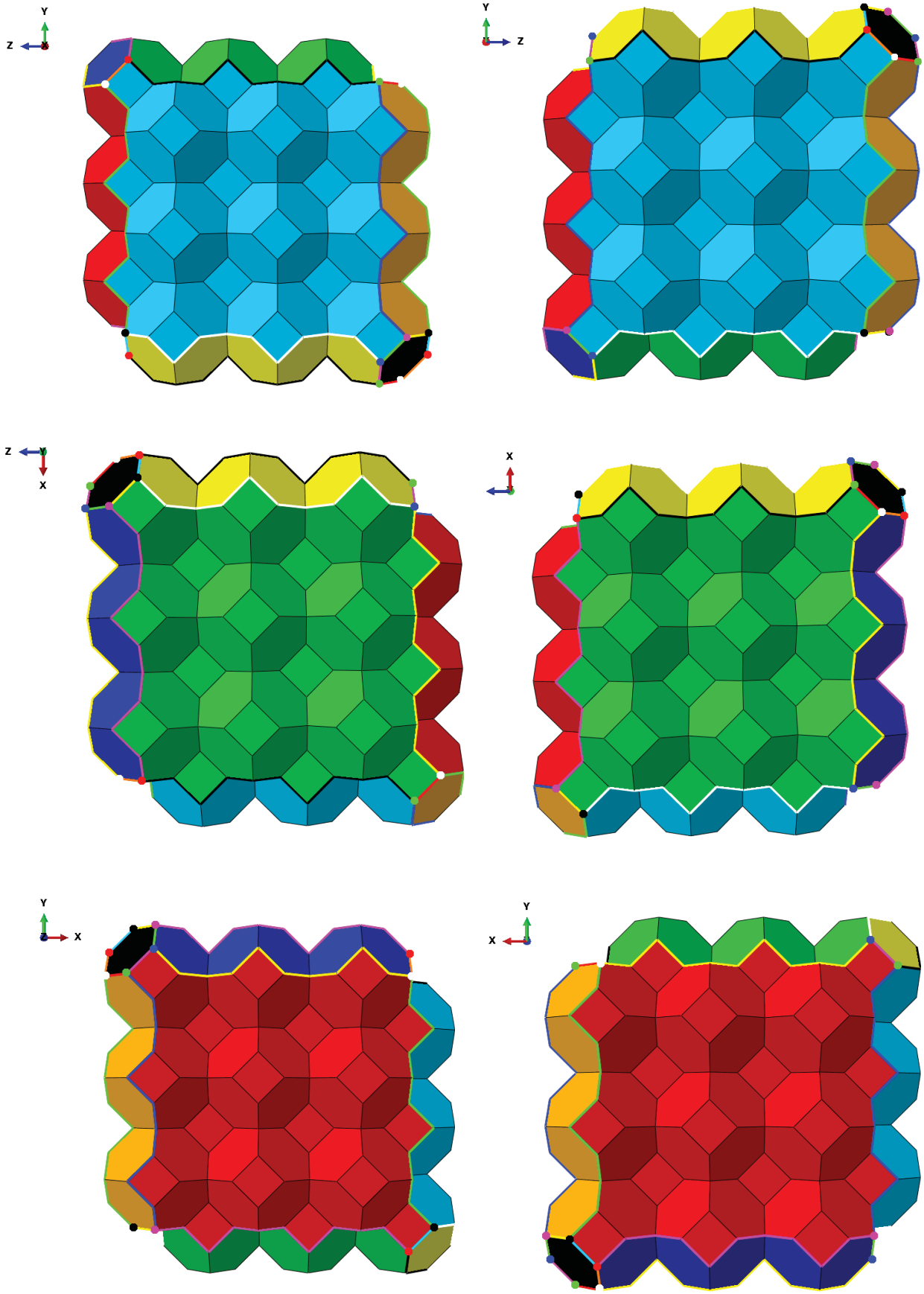

Figure A.3.: Normal projection views of the periodic cell. To achieve the given views the picture in the upper left side is rotated around X,Y and Z respectively. Top: Front- and back side, Mid: Top and bottom side, Bottom: Left and right side.

B. Abaqus related Issues

B.1. Consistent Units at a Nanoscale

Quantity	**SI**	**SI(nm)**
Length	m	nm
Force	N	N
Mass	kg	10^9 kg
Time	s	s
Stress	Pa = N/m^2	10^{18} Pa = N/nm^2
Energy	J	nJ = 10^{-9} J
Density	kg/m^3	10^9 kg/nm^3
Strain-energy-density	J/m^3	nJ/nm^3 = 10^{-12} J/cm^3 = 10^{-12} MPa

B.2. Abaqus Object Model

Figure B.1 exemplarily shows object trees of an .mdb and an .odb file respectively and the partition of them into objects and containers. According to the Abaqus scripting users manual a Container is an object that contains objects of a similar type. A container in the Abaqus object model can be either a repository or a sequence. For example, the "step" container in Figure B.1 is a repository that contains all the steps in the analysis and a script uses the "step" container to access these steps. A good idea is to use a variable to refer to an object, making scripts easier to read and understand.

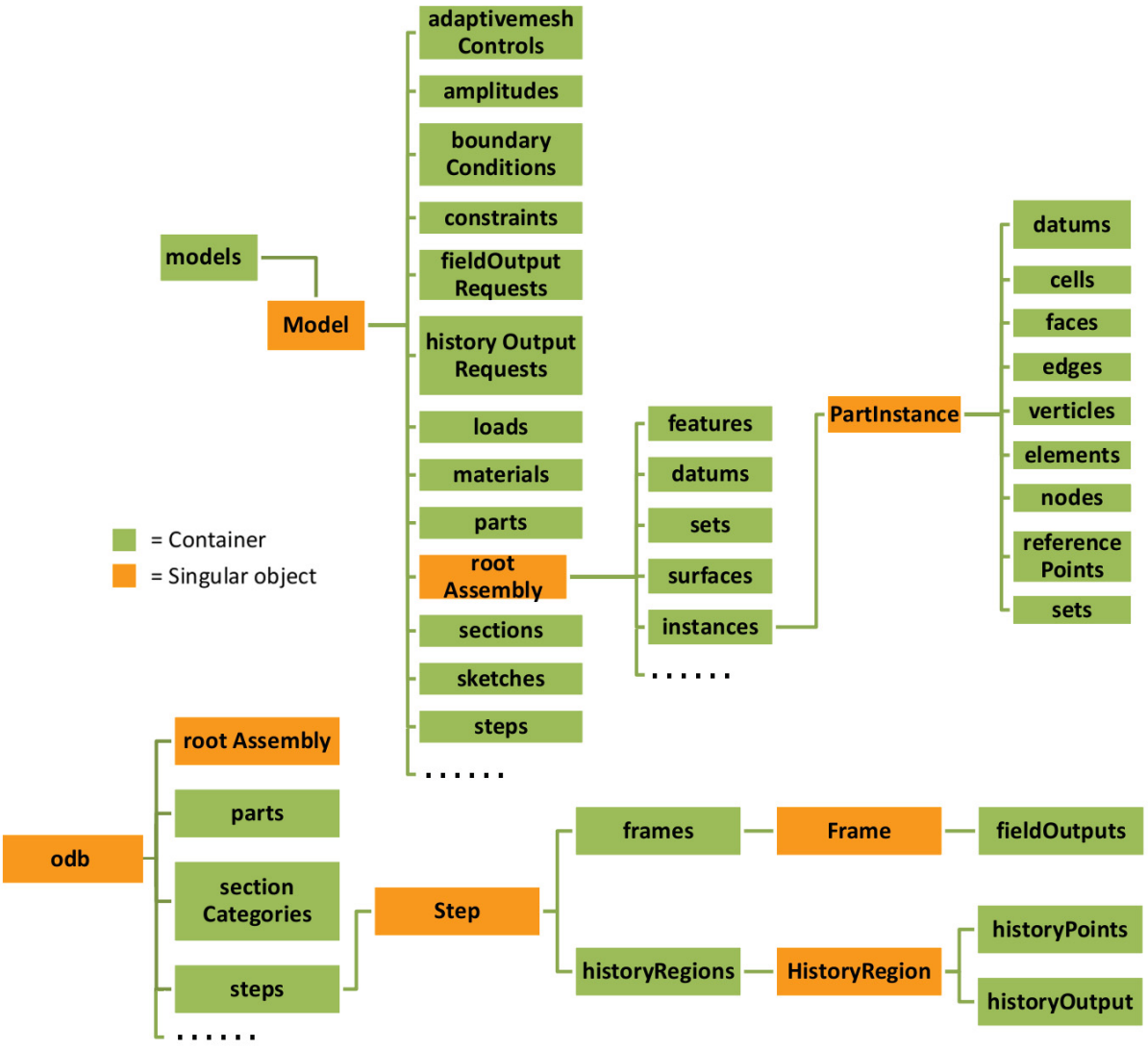

Figure B.1.: abaqus hierarchy same result with: frame number = -1 ... last frame, output property for instance S..Stress, SENER...Strainenergydensity etc.

B.3. Resource Optimization

The automated calculations are run on a "node" of a computer cluster. A node has 12 cores, each with a slightly more than 10 Gigabyte (Gb) Random Access Memor (RAM) resulting in a total amount of 128 Gb RAM and 2 Central Processing Units (CPUs) (E5-2667, 6-Core, 2.9/3.5 GHz, 15 Megabyte (MB) Cache). First a full transformation was simulated with a coarser mesh, following the transformation algorithm in subsection 3.5.2 to determine the sequence of grain-laminate pairs the martensite grains transform to incrementally minimize the total free energy. Obviously, the fastest way to do so is (i) to calculate as many states simultaneously as possible while (ii) ensuring that each job runs with enough resources, i.e. RAM and cores. For optimization purposes, first the speedup of one job was determined as a function of the parallelization on more than one core. As can be seen in Figure B.2 the additional speedup of one job reduces significantly with every additional core its parallelized on and even stagnates from 6 to eight cores. Regarding memory availability, each job of coarse mesh needs about 5-6 Gb of RAM. Another aspect is that Abaqus has a strict token policy, meaning that for each calculation using the Abaqus standart, explicit or CFD solver a certain amount of tokens is required as a function of the number of cores. This function is shown in Figure B.3 [71]. As a consequence of the two issues above in the full transformation 12 jobs are run simultaneously on one core respectively requiring 60 tokens, whereas for exact results assuring mesh convergence each determined increment was recalculated with a refined mesh using only two nodes but reserving all twelve nodes to reserve enough memory.

$$Tokens = \lfloor 5 \cdot Cores^{0.422} \rfloor$$

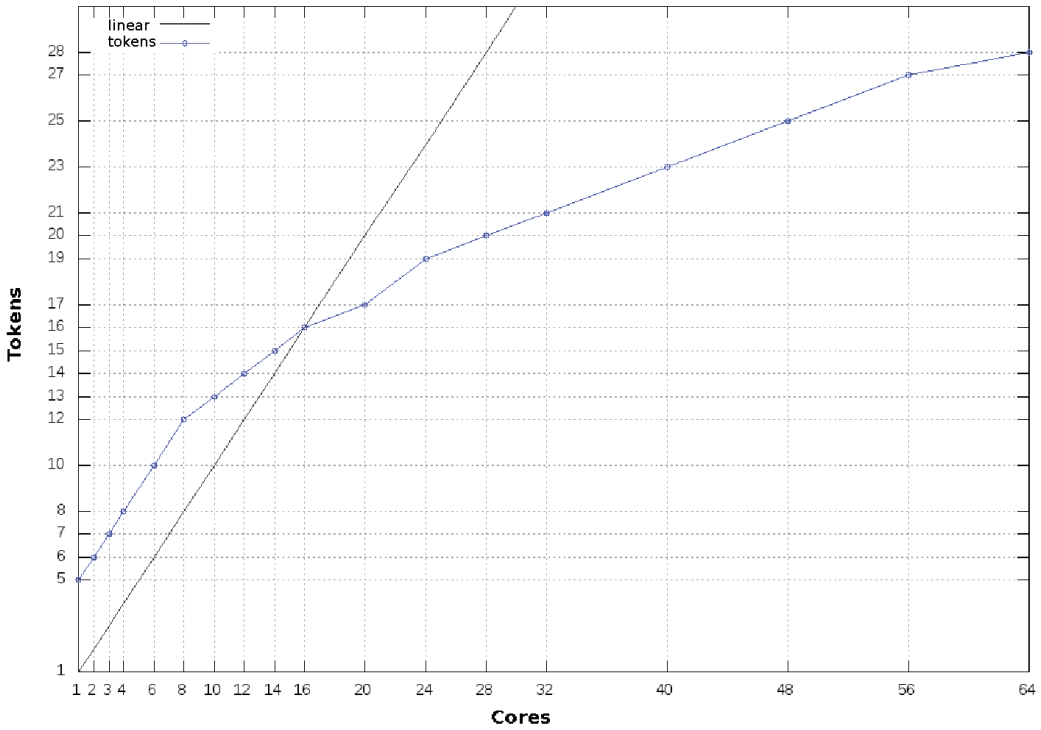

Figure B.3.: cores vs tokens

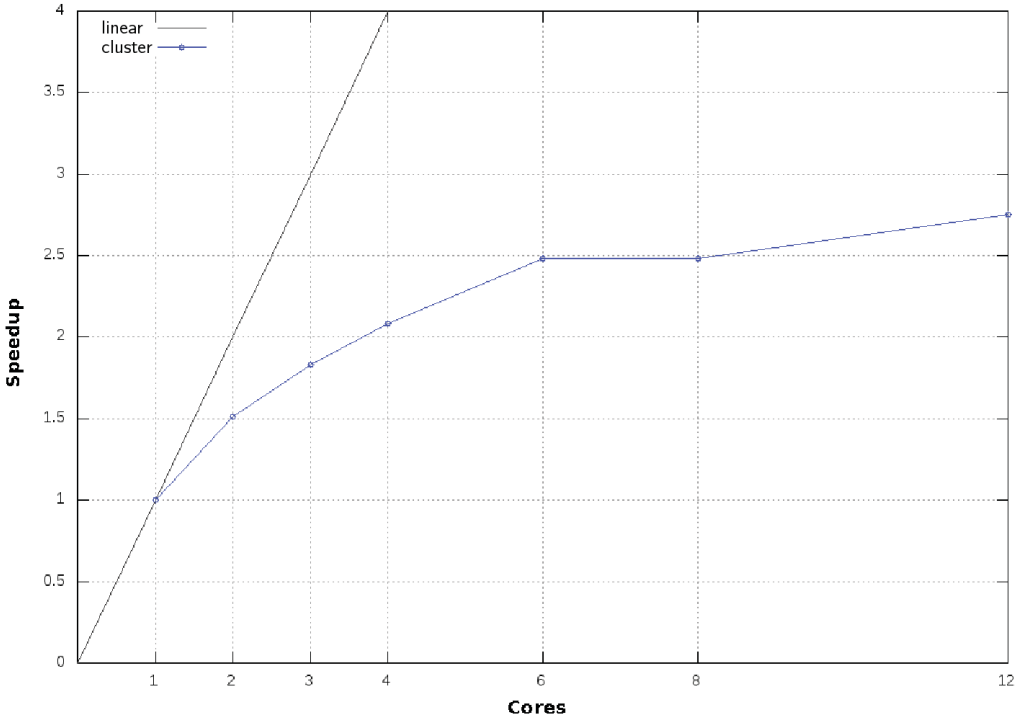

Figure B.2.: Visualization of the calculation time ratios (speedup) in the case of calculating one job parallelized on more than one core.

Finally it is mentioned that abaqus can work on the scratch disk space available on clusters normally having the advantage of faster I/O, leading to an increase in overall speed of execution. Here a temporary file system (tmpfs) [72], that is an improved replacement of the RAM-Disk common in Unix distributions, was used instead of the normal scratch space.

C. Static Input File Sections

C.1. Material and Job Data

```
*Material, name=austenite
*Elastic, type=ISOTROPIC
6.5e-8, 0.4
**
*Material, name=laminate1
*Elastic, type=ANISOTROPIC
2.865e-07,  1.275e-07,   1.80e-07,   1.00e-07,   1,275e-07,   2.865e-07,   0.,         0.
0.,         4.8e-08,     0.,         0.,         0.,          0.,          5,75e-08,   0.
0.,         0.,          0.,         0.,         4,8e-08
*Expansion, type=ANISO
0.0234,     -0.0381,     0.0234,     0.,         -0.1186,     0.
**
*Material, name=laminate2
*Elastic, type=ANISOTROPIC
2.865e-07,  1.005e-07,   2.86e-07,   1.275e-07,  1,275e-07,   1,80e-07,    0.,         0.
0.,         5.75e-08,    0.,         0.,         0.,          0.,          4.8e-08,    0.
0.,         0.,          0.,         0.,         4.8e-8
*Expansion, type=ANISO
0.0234,     0.0234,      -0.0381,    0.1186,     0.,      0.

*Material, name=laminate3
*Elastic, type=ANISOTROPIC
2.865e-07,  1.275e-07,   1.80e-07,   1.005e-07,  1.275e-07,   2.865e-07,   0.,         0.
0.,         4.8e-08,     0.,         0.,         0.,          0.,          5.75e-08,   0.
0.,         0.,          0.,         0.,         4.8e-08
*Expansion, type=ANISO
0.0234,     -0.0381,      0.0234,    0., 0.1186,      0.
**
*Material, name=laminate4
*Elastic, type=ANISOTROPIC
2.865e-07,  1.005e-07,   2.865e-07,  1.275e-07,  1.275e-07,   1.80e-07,    0.,         0.
0.,         5.75e-08,    0.,         0.,         0.,          0.,          4.8e-08,    0.
0.,         0.,          0.,         0.,         4.8e-08
*Expansion, type=ANISO
```

```
0.0234,     0.0234,     -0.0381,    -0.1186,     0.,      0.
**
*Material, name=laminate5
*Elastic, type=ANISOTROPIC
1.80e-07,   1.275e-07,   2.865e-07,  1.275e-07,  1.005e-07,   2.865e-07,   0.,         0.
0.,         4.8e-08,     0.,         0.,         0.,         0.,           4.8e-8,     0.
0.,         0.,          0.,         0.,         5.75e-08
*Expansion, type=ANISO
-0.0381,    0.0234,      0.0234,     0., 0.,        -0.1186
**
*Material, name=laminate6
*Elastic, type=ANISOTROPIC
1.80e-07,   1.275e-07,   2.865e-07,  1.275e-07,  1.005e-07,   2.865e-07,    0.,        0.
0.,         4.8e-08,     0.,         0.,         0.,          0.,           4.8e-08,   0.
0.,         0.,          0.,         0.,         5.75e-08
*Expansion, type=ANISO
-0.0381,    0.0234,      0.0234,     0., 0.,         0.1186
**
*Initial Conditions, type=TEMPERATURE
ALLNODES, 0.
**
*Step, name=Transformation
MartensiticTransformation_applied
*Static
1., 1., 1e-05, 1.
**
*Temperature
ALLNODES, 1.
**
*BOUNDARY
bc1,        1, 3
bc2,        1, 2
bc3,        1, 1
**
*Restart, write, frequency=0
**
*Output, field
*Element Output, directions=YES
E, S, SENER, IVOL
**
*Node Output
U
**
*Output, history, variable=PRESELECT
*Energy Output
ALLIE
*Energy Print
*End Step
```

C.2. Periodic Boundary Equations

```
*EQUATION
3
faceFront, 1, 1., faceBack, 1, -1., REFD, 1, -1.,
3
faceFront, 2, 1., faceBack, 2, -1., REFS, 1, -1.,
3
faceFront, 3, 1., faceBack, 3, -1., REFS, 3, -1.,
3
faceTop, 1, 1., faceBot, 1, -1., REFS, 1, -1.,
3
faceTop, 2, 1., faceBot, 2, -1., REFD, 2, -1.,
3
faceTop, 3, 1., faceBot, 3, -1., REFS, 2, -1.,
3
faceRight, 1, 1., faceLeft, 1, -1., REFS, 3, 1.,
3
faceRight, 2, 1., faceLeft, 2, -1., REFS, 2, 1.,
3
faceRight, 3, 1., faceLeft, 3, -1., REFD, 3, 1.,
4
face_edge_top_left, 1, 1., face_edge_bot_right, 1, -1., REFS, 1, -1., REFS, 3, -1.,
4
face_edge_top_left, 2, 1., face_edge_bot_right, 2, -1., REFD, 2, -1., REFS, 2, -1.,
4
face_edge_top_left, 3, 1., face_edge_bot_right, 3, -1., REFS, 2, -1., REFD, 3, -1.,
4
face_edge_front_right, 1, 1., face_edge_back_left, 1, -1., REFD, 1, -1., REFS, 1, 1.,
4
face_edge_front_right, 2, 1., face_edge_back_left, 2, -1., REFS, 1, -1., REFS, 2, 1.,
4
face_edge_front_right, 3, 1., face_edge_back_left, 3, -1., REFS, 3, -1., REFD, 3, 1.,
4
face_edge_front_bot, 1, 1., face_edge_back_top, 1, -1., REFD, 1, -1., REFS, 1, 1.,
4
face_edge_front_bot, 2, 1., face_edge_back_top, 2, -1., REFS, 1, -1., REFD, 2, 1.,
4
face_edge_front_bot, 3, 1., face_edge_back_top, 1, -1., REFS, 3, -1., REFS, 2, 1.,
5
face_edge_front_bot_solo, 1, 1., face_edge_back_top_solo, 1, -1., REFD, 1, -1., REFS, 1, 1.,
REFS, 3, 1.,
5
face_edge_front_bot_solo, 2, 1., face_edge_back_top_solo, 2, -1., REFS, 1, -1., REFD, 2, 1.,
REFS, 2, 1.,
5
face_edge_front_bot_solo, 3, 1., face_edge_back_top_solo, 3, -1., REFS, 3, -1., REFS, 2, 1.,
REFD, 3, 1.,
3
edge_back_top, 1, 1., edge_front_top, 1, -1., REFD, 1, 1.,
3
edge_back_top, 2, 1., edge_front_top, 2, -1., REFS, 1, 1.,
3
edge_back_top, 3, 1., edge_front_top, 3, -1., REFS, 3, 1.,
3
edge_front_bot_bot, 1, 1., edge_front_top, 1, -1., REFS, 1, 1.,
3
edge_front_bot_bot, 2, 1., edge_front_top, 2, -1., REFD, 2, 1.,
3
edge_front_bot_bot, 3, 1., edge_front_top, 3, -1., REFS, 2, 1.,
```

```
3
edge_front_bot, 1, 1., edge_back_bot, 1, -1., REFD, 1, -1.,
3
edge_front_bot, 2, 1., edge_back_bot, 2, -1., REFS, 1, -1.,
3
edge_front_bot, 3, 1., edge_back_bot, 3, -1., REFS, 3, -1.,
3
edge_back_top_top, 1, 1., edge_back_bot, 1, -1., REFS, 1, -1.,
3
edge_back_top_top, 2, 1., edge_back_bot, 2, -1., REFD, 2, -1.,
3
edge_back_top_top, 3, 1., edge_back_bot, 3, -1., REFS, 2, -1.,
3
edge_front_right_right, 1, 1., edge_front_left, 1, -1., REFS, 3, 1.,
3
edge_front_right_right, 2, 1., edge_front_left, 2, -1., REFS, 2, 1.,
3
edge_front_right_right, 3, 1., edge_front_left, 3, -1., REFD, 3, 1.,
3
edge_back_left, 1, 1., edge_front_left, 1, -1., REFD, 1, 1.,
3
edge_back_left, 2, 1., edge_front_left, 2, -1., REFS, 1, 1.,
3
edge_back_left, 3, 1., edge_front_left, 3, -1., REFS, 3, 1.,
3
edge_front_right, 1, 1., edge_back_right, 1, -1., REFD, 1, -1.,
3
edge_front_right, 2, 1., edge_back_right, 2, -1., REFS, 1, -1.,
3
edge_front_right, 3, 1., edge_back_right, 3, -1., REFS, 3, -1.,
3
edge_back_left_left, 1, 1., edge_back_right, 1, -1., REFS, 3, -1.,
3
edge_back_left_left, 2, 1., edge_back_right, 2, -1., REFS, 2, -1.,
3
edge_back_left_left, 3, 1., edge_back_right, 3, -1., REFD, 3, -1.,
3
edge_1_solo1, 1, 1., edge_solo_front_top_right, 1, -1., REFS, 1, 1.,
3
edge_1_solo1, 2, 1., edge_solo_front_top_right, 2, -1., REFD, 2, 1.,
3
edge_1_solo1, 3, 1., edge_solo_front_top_right, 3, -1., REFS, 2, 1.,
4
edge_1_solo2, 1, 1., edge_solo_front_top_right, 1, -1., REFD, 1, 1., REFS, 3, -1.,
4
edge_1_solo2, 2, 1., edge_solo_front_top_right, 2, -1., REFS, 2, 1., REFS, 2, -1.,
4
edge_1_solo2, 3, 1., edge_solo_front_top_right, 3, -1., REFS, 3, 1., REFD, 3, -1.,
4
edge_2_solo1, 1, 1., edge_solo_front_top_left, 1, -1., REFS, 1, 1., REFS, 3, 1.,
4
edge_2_solo1, 2, 1., edge_solo_front_top_left, 2, -1., REFD, 2, 1., REFS, 2, 1.,
4
edge_2_solo1, 3, 1., edge_solo_front_top_left, 3, -1., REFS, 2, 1., REFD, 3, 1.,
3
edge_2_solo2, 1, 1., edge_solo_front_top_left, 1, -1., REFD, 1, 1.,
3
edge_2_solo2, 2, 1., edge_solo_front_top_left, 2, -1., REFS, 1, 1.,
3
edge_2_solo2, 3, 1., edge_solo_front_top_left, 3, -1., REFS, 3, 1.,
3
edge_3_solo1, 1, 1., edge_solo_front_bot_left, 1, -1., REFS, 3, 1.,
```

```
3
edge_3_solo1, 2, 1., edge_solo_front_bot_left, 2, -1., REFS, 2, 1.,
3
edge_3_solo1, 3, 1., edge_solo_front_bot_left, 3, -1., REFD, 3, 1.,
4
edge_3_solo2, 1, 1., edge_solo_front_bot_left, 1, -1., REFD, 1, 1., REFS, 1, -1.,
4
edge_3_solo2, 2, 1., edge_solo_front_bot_left, 2, -1., REFS, 1, 1., REFD, 2, -1.,
4
edge_3_solo2, 3, 1., edge_solo_front_bot_left, 3, -1., REFS, 3, 1., REFS, 2, -1.,
4
edge_4_solo1, 1, 1., edge_solo_back_bot_left, 1, -1., REFD, 1, -1., REFS, 3, 1.,
4
edge_4_solo1, 2, 1., edge_solo_back_bot_left, 2, -1., REFS, 1, -1., REFS, 2, 1.,
4
edge_4_solo1, 3, 1., edge_solo_back_bot_left, 3, -1., REFS, 3, -1., REFD, 3, 1.,
3
edge_4_solo2, 1, 1., edge_solo_back_bot_left, 1, -1., REFS, 1, -1.,
3
edge_4_solo2, 2, 1., edge_solo_back_bot_left, 2, -1., REFD, 2, -1.,
3
edge_4_solo2, 3, 1., edge_solo_back_bot_left, 3, -1., REFS, 2, -1.,
3
edge_5_solo1, 1, 1., edge_solo_back_bot_right, 1, -1., REFD, 1, -1.,
3
edge_5_solo1, 2, 1., edge_solo_back_bot_right, 2, -1., REFS, 1, -1.,
3
edge_5_solo1, 3, 1., edge_solo_back_bot_right, 3, -1., REFS, 3, -1.,
4
edge_5_solo2, 1, 1., edge_solo_back_bot_right, 1, -1., REFS, 1, -1., REFS, 3, -1.,
4
edge_5_solo2, 2, 1., edge_solo_back_bot_right, 2, -1., REFD, 2, -1., REFS, 2, -1.,
4
edge_5_solo2, 3, 1., edge_solo_back_bot_right, 3, -1., REFS, 2, -1., REFD, 3, -1.,
4
edge_6_solo1, 1, 1., edge_solo_back_top_right, 1, -1., REFD, 1, -1., REFS, 1, 1.,
4
edge_6_solo1, 2, 1., edge_solo_back_top_right, 2, -1., REFS, 1, -1., REFD, 2, 1.,
4
edge_6_solo1, 3, 1., edge_solo_back_top_right, 3, -1., REFS, 3, -1., REFS, 2, 1.,
3
edge_6_solo2, 1, 1., edge_solo_back_top_right, 1, -1., REFS, 3, -1.,
3
edge_6_solo2, 2, 1., edge_solo_back_top_right, 2, -1., REFS, 2, -1.,
3
edge_6_solo2, 3, 1., edge_solo_back_top_right, 3, -1., REFD, 3, -1.,
3
edge_left_top_top, 1, 1., edge_left_bot, 1, -1., REFS, 1, -1.,
3
edge_left_top_top, 2, 1., edge_left_bot, 2, -1., REFD, 2, -1.,
3
edge_left_top_top, 3, 1., edge_left_bot, 3, -1., REFS, 2, -1.,
3
edge_right_bot, 1, 1., edge_left_bot, 1, -1., REFS, 3, 1.,
3
edge_right_bot, 2, 1., edge_left_bot, 2, -1., REFS, 2, 1.,
3
edge_right_bot, 3, 1., edge_left_bot, 3, -1., REFD, 3, 1.,
3
edge_left_top, 1, 1., edge_right_top, 1, -1., REFS, 3, -1.,
3
edge_left_top, 2, 1., edge_right_top, 2, -1., REFS, 2, -1.,
```

```
3
edge_left_top, 3, 1., edge_right_top, 3, -1., REFD, 3, -1.,
3
edge_right_bot_bot, 1, 1., edge_right_top, 1, -1., REFS, 1, 1.,
3
edge_right_bot_bot, 2, 1., edge_right_top, 2, -1., REFD, 2, 1.,
3
edge_right_bot_bot, 3, 1., edge_right_top, 3, -1., REFS, 2, 1.,
4
c1, 1, 1., c_f_t_l_2, 1, -1., REFS, 1, 1., REFS, 3, 1.,
4
c1, 2, 1., c_f_t_l_2, 2, -1., REFD, 2, 1., REFS, 2, 1.,
4
c1, 3, 1., c_f_t_l_2, 3, -1., REFS, 2, 1., REFD, 3, 1.,
3
c11, 1, 1., c_f_t_l_2, 1, -1., REFD, 1, 1.,
3
c11, 2, 1., c_f_t_l_2, 2, -1., REFS, 1, 1.,
3
c11, 3, 1., c_f_t_l_2, 3, -1., REFS, 3, 1.,
3
c_f_t_r_2, 1, 1., c_f_t_l_2, 1, -1., REFS, 3, 1.,
3
c_f_t_r_2, 2, 1., c_f_t_l_2, 2, -1., REFS, 2, 1.,
3
c_f_t_r_2, 3, 1., c_f_t_l_2, 3, -1., REFD, 3, 1.,
4
c2, 1, 1., c_f_t_l_1, 1, -1., REFS, 1, 1., REFS, 3, 1.,
4
c2, 2, 1., c_f_t_l_1, 2, -1., REFD, 2, 1., REFS, 2, 1.,
4
c2, 3, 1., c_f_t_l_1, 3, -1., REFS, 2, 1., REFD, 3, 1.,
3
c22, 1, 1., c_f_t_l_1, 1, -1., REFD, 1, 1.,
3
c22, 2, 1., c_f_t_l_1, 2, -1., REFS, 1, 1.,
3
c22, 3, 1., c_f_t_l_1, 3, -1., REFS, 3, 1.,
3
c_f_bo_l_2, 1, 1., c_f_t_l_1, 1, -1., REFS, 1, 1.,
3
c_f_bo_l_2, 2, 1., c_f_t_l_1, 2, -1., REFD, 2, 1.,
3
c_f_bo_l_2, 3, 1., c_f_t_l_1, 3, -1., REFS, 2, 1.,
3
c3, 1, 1., c_f_bo_l_1, 1, -1., REFS, 3, 1.,
3
c3, 2, 1., c_f_bo_l_1, 2, -1., REFS, 2, 1.,
3
c3, 3, 1., c_f_bo_l_1, 3, -1., REFD, 3, 1.,
4
c33, 1, 1., c_f_bo_l_1, 1, -1., REFD, 1, 1., REFS, 1, -1.,
4
c33, 2, 1., c_f_bo_l_1, 2, -1., REFS, 1, 1., REFD, 2, -1.,
4
c33, 3, 1., c_f_bo_l_1, 3, -1., REFS, 3, 1., REFS, 2, -1.,
3
c_ba_bo_l_2, 1, 1., c_f_bo_l_1, 1, -1., REFD, 1, 1.,
3
c_ba_bo_l_2, 2, 1., c_f_bo_l_1, 2, -1., REFS, 1, 1.,
3
c_ba_bo_l_2, 3, 1., c_f_bo_l_1, 3, -1., REFS, 3, 1.,
```

```
3
c4, 1, 1., c_ba_bo_r_1, 1, -1., REFD, 1, -1.,
3
c4, 2, 1., c_ba_bo_r_1, 2, -1., REFS, 1, -1.,
3
c4, 3, 1., c_ba_bo_r_1, 3, -1., REFS, 3, -1.,
4
c44, 1, 1., c_ba_bo_r_1, 1, -1., REFS, 1, -1., REFS, 3, -1.,
4
c44, 2, 1., c_ba_bo_r_1, 2, -1., REFD, 2, -1., REFS, 2, -1.,
4
c44, 3, 1., c_ba_bo_r_1, 3, -1., REFS, 2, -1., REFD, 3, -1.,
3
c_ba_bo_l_1, 1, 1., c_ba_bo_r_1, 1, -1., REFS, 3, -1.,
3
c_ba_bo_l_1, 2, 1., c_ba_bo_r_1, 2, -1., REFS, 2, -1.,
3
c_ba_bo_l_1, 3, 1., c_ba_bo_r_1, 3, -1., REFD, 3, -1.,
3
c5, 1, 1., c_ba_bo_r_2, 1, -1., REFD, 1, -1.,
3
c5, 2, 1., c_ba_bo_r_2, 2, -1., REFS, 1, -1.,
3
c5, 3, 1., c_ba_bo_r_2, 3, -1., REFS, 3, -1.,
4
c55, 1, 1., c_ba_bo_r_2, 1, -1., REFS, 1, -1., REFS, 3, -1.,
4
c55, 2, 1., c_ba_bo_r_2, 2, -1., REFD, 2, -1., REFS, 2, -1.,
4
c55, 3, 1., c_ba_bo_r_2, 3, -1., REFS, 2, -1., REFD, 3, -1.,
3
c_ba_t_r_1, 1, 1., c_ba_bo_r_2, 1, -1., REFS, 1, -1.,
3
c_ba_t_r_1, 2, 1., c_ba_bo_r_2, 2, -1., REFD, 2, -1.,
3
c_ba_t_r_1, 3, 1., c_ba_bo_r_2, 3, -1., REFS, 2, -1.,
4
c6, 1, 1., c_ba_t_r_2, 1, -1., REFD, 1, -1., REFS, 1, 1.,
4
c6, 2, 1., c_ba_t_r_2, 2, -1., REFS, 1, -1., REFD, 2, 1.,
4
c6, 3, 1., c_ba_t_r_2, 3, -1., REFS, 3, -1., REFS, 2, 1.,
3
c66, 1, 1., c_ba_t_r_2, 1, -1., REFS, 3, -1.,
3
c66, 2, 1., c_ba_t_r_2, 2, -1., REFS, 2, -1.,
3
c66, 3, 1., c_ba_t_r_2, 3, -1., REFD, 3, -1.,
3
c_f_t_r_1, 1, 1., c_ba_t_r_2, 1, -1., REFD, 1, -1.,
3
c_f_t_r_1, 2, 1., c_ba_t_r_2, 2, -1., REFS, 1, -1.,
3
c_f_t_r_1, 3, 1., c_ba_t_r_2, 3, -1., REFS, 3, -1.,
```

D. Neper Run Parameters

```
# Generate tesselation

neper -T   # use the tesselation module
-n 888    # specifiy the number of grains
-statpoly id,body,vol,area  # create a statistic file and write the specified variables to it
-id 1    # specify identifier of tesselation
-o n888    # specify output name
-sort poly   # name the created nodesets in ascending order
-scale 650:650:650  # specify scaling factors for x:y:z (default 1:1:1)
-morpho equiaxed   # alternatively columnar (+direction) or lamellae


# Mesh tesselation with the specified mesh

neper -M   # use the meshing module
n888.tess   # specify tesselation filename
-o n888_mesh   # specify the output filename
-format inp   # specify the output as abaqus input-file
-order 2   # specify the element order
# specify the relative characteristic length for the inner and outer mesh
-rcl "0.5,body<2:3"
# refine mesh as far as possible using multiple meshing algorythms and use the best result
-mesh3dalgo netg:gmsh,netg:netg,netg:gmne -nset facebodies -faset faces
```

E. Python Scripts

E.1. transEnergymin.py

```
''' This script evaluates the grain-laminate pair which minimizes the total free energy
density of the specified RVE upon transformation. If previous increments are fully
calculated the script automatically starts from the last state that is saved in file
'continuing_data'. The script needs N = (1 + maxGrainNr)* 6 * (maxGrainNr / 2) calculations
to find the energy minimizing state of a full transformation. A preselection of more likely
states based on previous results can be done reducing N significantly. At the beginning
the script-parameters and the used textfiles have to be specified.'''

# python modules
import cPickle as pickle  # Phython module to save intermediate results conveniently
import shutil    # high level file operations like copying
import glob    # Unix style pathname pattern expansion
import os    # miscellaneous operating system interfaces
import time    # module for time access
# my modules
import write
import automate
import material


#-----< SPECIFY SCRIPT-PARAMETERS >-----------------------------------------------------#
pbc = True # use the specified periodic boundary equations or a self-consistent matrix
preselection = True # define if presection is used
# define at which increments (number of transformed grains) all possibillities for the
# preselection are calculated
selected_steps = [1,15,30,45,60,75,85,95,101,107,113,117,121,124]
# set the timeout after which a single calculation is killed if it has not finished
timeout = 1200
# initialize array of numbers that define the transformed material behavior (here laminates)
laminateVariants = [1,2,3,4,5,6 ]
# choose between periodic boundary conditions for the regular tesselation or the self
# consistent matrix for the random microstrocture and specify the required files
if pbc == True:
  material_jobData_filename = ' path/to/file' # holds the equations for the PBCs
  geometry_filename = ' path/to/file' # fixed mesh and orientations for the PBC model
  totalGrainAmount = 128 # total number of equally sized octahedra in the RVE
  grainVolume = 268000.0 # Volume of a sphere of 80nm diameter
else:
  material_jobData_filename = ' path/to/file' # static inputfile section of random RVE
  geometry_filename = ' path/to/file' # fixed mesh for the ESCM
  # here the orientations are written explicitly since they are also used for the
  # averaging of the material properties in each increment
  orientation_filename = ' path/to/file'
```

```
#-----< DETERMINE LAST STATE of simulation (if) or PREPARE CALCULATION (else) >----------#
if os.path.isfile('saves/continuing_data'): # check if there are calculation results
  # open file and load results in the same order they were written
  cont = open('saves/continuing_data','rb')
  martensiteAmount = pickle.load( cont )
  oris = pickle.load( cont ) if pbc == False else 0 # ternary operator assignment
  martensiteGrains = pickle.load( cont )
  austeniteGrains = pickle.load( cont )
  chemical_drivingForce = pickle.load( cont )
  total_strainEner_cell_before = pickle.load( cont )
  selected_variants = pickle.load( cont )
  cont.close()

  # get amount of grains for the randomly generated microstructure
  totalGrainAmount = len(austeniteGrains) + len(martensiteGrains)
  # Define name of .odb file containing latest evaluated result
  odbname = 'saves/Outputfile_'+ str( martensiteAmount - 1 )+ '_'+ \
    str( martensiteGrains[-1][0] ) + '_'+ str( martensiteGrains[-1][1] )+'.odb'
  #
else: # create save directory and files
  martensiteGrains = selected_variants = []
  martensiteAmount = 1
  os.system('mkdir saves') # create directory where results are saved
  with open('saves/save_grain', 'w') as save_grain:
    save_grain.write('grainNr\tlaminateNr\tgrainVol\tdragEner_spec\t\t' + \
      'delta_totStrainEner_spec\tdelta_allEnergies\ttransformingEnergy\n')
  with open('saves/save_model', 'w') as save_model:
    save_model.write('tot_strainEner\t\tdelta_tot_strainEner\ttot_aveSener' + \
      '\t\tivol_aust\t\tivol_mart\t\tave_sener_aust\t\tave_sener_mart\n')
  #
  if pbc == False:
    odbname = exodb_filename
    austeniteGrains = automate.get_volumes_and_laminates( exodb_filename )
    # austeniteGrains [grainNr, grainvolume, grainmaterial]
    totalGrainAmount = len(austeniteGrains) # get Nr of grains in the ESCM
  else:
    austeniteGrains = []
    for i in range(totalGrainAmount):
      austeniteGrains.append( [i+1, grainVolume, 0] )
      # recall that the grain volume is equal for all octahedra


#-----< CALCULATE SELF CONSISTENT MATERIAL PROPERTIES and PRESELECT more likely states >--#
if pbc == False:
  # list [grainnumber, grainvolume, grainmaterial]
  # necessary for averaging anisotropic data
  graindata, Vinner = automate.get_volumes_and_laminates.get( odbname )
  #
  # calculate the averaged material properties from the last energy-minimizing state
  C_ave = material.selfconsistent_matrix( oris, graindata, Vinner )
  del graindata, Vinner
  #
# Limit number of calculations by preselecting more likely states
if preselection == True:
  # calculate all possible states only in every selected stepwidth
  if martensiteAmount in selected_steps:
    preselection = False
  if (martensiteAmount-1) in selected_steps:
    selected_variants = automate.preselect( totalGrainAmount, martensiteAmount )
```

```
#-----< INPUTFILE CREATION >-----------------------------------------------------------------#
# All possible or preselected states of one more transformed grain are evaluated
for austeniteGrain in austeniteGrains:
  # Every not transformed grain can transform in multiple ways
  for laminate in laminateVariants:
    if ( [austeniteGrain[0], laminate] in selected_variants) or preselection == False:
      write.writeInputfile( martensiteAmount, austeniteGrain, \
        austeniteGrains, martensiteGrains, laminate, pbc, \
        geometry_filename, material_jobData_filename, C_ave )



#-----< JOB SUBMISSION of all Jobs that were created >---------------------------------#
automate.submitjobs( austeniteGrains, martensiteAmount, laminateVariants, preselection, \
                     selected_variants, timeout )
# delay to finish operations on the .odb files so that no *lck files are created
time.sleep(60)


#-----< EVALUATE ALL jobs and SET PARAMETERS for the transformation of the next grain >---#
if martensiteAmount == 1:
  evaluationData = automate.findMinimumEnergy( austeniteGrains, martensiteAmount, \
                                               laminateVariants, preselection)
  # evaluationData = [0-delta_G, 1-GrainNr, 2-GrainLaminate, 3-GrainVol,
  # 4-dragEner_spec, 5-delta_totalStrain_spec, 6-total_strainEner_cell]
  chemical_drivingForce = max( evaluationData )[0] # note that this is a negative value
else:
  evaluationData = automate.findMinimumEnergy(austeniteGrains, martensiteAmount,\
                                      laminateVariants, preselection, selected_variants, \
                                      total_strainEner_cell_before, chemical_drivingForce)
# # the optimum grain is that with minimum delta_G
foundGrain = min( abs( evaluationData ) )
# the max function acts on the first entry which is 'delta_G'
#
# if delta_G reaches a new negative maximum the chemical driving force
# has to be increased for further transformations
if foundGrain[0] < 0: # if delta_G < 0
  chemical_drivingForce = foundGrain[0]
#
total_strainEner_cell_before = foundGrain[6]


#-----< WRITE DATA of all runs and energy-minimizing configuration to files >-------------#
write.writeSaves( martensiteAmount, evaluationData, chemical_drivingForce, foundGrain )
del evaluationData


#-----< MOVE FOUNDGRAIN from austeniteGrains to martensiteGrains >-----------------------#
for index, iGrain in enumerate(austeniteGrains):
  # remove found grain from austeniteGrains
  if iGrain[0] == foundGrain[1]:
    austeniteGrains.pop( index )
# add found grain - material pair to martensiteGrains
martensiteGrains.append( [ foundGrain[1], foundGrain[2] ]
)
#
# remove foundgrain from selected_variants if preselection is used
if preselection == True:
  for index, iGrain in enumerate( selected_variants ):
    if iGrain[0] == foundGrain[1]:
      selected_variants.pop( index )
```

```
#-----< SAVE (PICKLE) EVALUATED NECESSARY VARIABLES for the next increment >--------------#
# it is crucial that the values are 'loaded' in the same order they are 'dumped'
cont = open('saves/continuing_data','wb')
pickle.dump( martensiteAmount + 1, cont )
if pbc == False: pickle.dump( oris, cont )
pickle.dump( martensiteGrains, cont )
pickle.dump( austeniteGrains, cont )
pickle.dump( chemical_drivingForce, cont )
pickle.dump( total_strainEner_cell_before, cont )
pickle.dump( selected_variants, cont )
cont.close()


#-----< SAVE FILES OF FOUNDGRAIND AND DELETE THE REST >----------------------------------#
savefilenames = glob.glob( '*_' + str( martensiteAmount ) +'_'+ str( foundGrain[1] ) + \
         '_'+ str( foundGrain[2] ) +'*') # example '*_17_44_5*'
for i in savefilenames:
  shutil.move(i,'saves') # generally: src --> destination, here: i --> saves
#
# delete all other files
os.system('rm *.*')


#-----< CREATE STOPPINGFILE >------------------------------------------------------------#
# Obviously Abaqus only allows a maximum number of around 1000 jobs in one interactive
# Python # session. As a workaround, the script is written for one increment and is
# recalled from an external bash script. When the transformation is finished, the file
# finish_loop file is created # causing the bash-script not to recall the script anymore
if martensiteAmount == totalGrainAmount:
  with open('saves/finish_loop','w') as f:
    pass
```

E.2. write.py

```
'''This module creates the specified inputfiles and writes simulation results to files '''

import shutil
from automate import evaluate_odb

def writeInputfile( martensiteAmount, austeniteGrain, austeniteGrains, martensiteGrains, \
    laminate, pbc, geometry_filename, material_jobData_filename, C_ave = 0 ):
  ''' creates an inputfile according to the specified parameters '''
  #
  # specify the name of the created inputfile
  inputFile_name = 'Inputfile_'+ str(martensiteAmount)+'_'+ str(austeniteGrain[0])+'_'+ \
  str(laminate) +'.inp'
  # the first section of the created inputfile is the used mesh from the specified
  # external file. Copy this external file and rename it to the specified inputfile name
  shutil.copy2( geometry_filename, inputFile_name )
  # append section definitions and assignments to inputfile according to previous results
  with open(inputFile_name, 'a') as ifile:
    # ----- write sections -----
    for iGrain in austeniteGrains:
      # only write currently transforming grain once
      if austeniteGrain == iGrain: continue
      string = '*Solid Section, elset=transig_' + str( iGrain[0]) + \
      ', orientation=Ori_'+ str(iGrain[0]) +', material=austenite\n'
      ifile.write(string)
```

```
    # write sections for already transformed grains
    for iGrain in martensiteGrains:
      # iGrain = [ grainNr, laminate ]
      string = '*Solid Section, elset=transig_'+ str( iGrain[0] ) + \
        ', orientation=Ori_'+ str( iGrain[0] ) + ', material=laminate'+ \
        str( iGrain[1] ) +'\n'
          ifile.write(string)
    # write section of additional grain transforming in this increment
    string = '*Solid Section, elset=transig_' + str( austeniteGrain[0] ) + \
      ', orientation=Ori_' + str( austeniteGrain[0] ) + ', material=laminate' + \
      str( laminate ) +'\n'
    ifile.write(string)
    # ---- write material and jobdata -----
    if pbc == False:
      # write section for self consistent matrix, an orientation is
      # needed because self consistent isotropic properties are given as
      # averaged anisotropic tensor
      string = '*Solid Section, elset=matrix, orientation=Ori_1,' + \
        'material=selfconsistentIsotropic\n'
      ifile.write(string)
      ifile.write('*Material, name=selfconsistentIsotropic\n*Elastic, type=ANISOTROPIC\n')
      # the specification due to abaqus is first and second line 8
      # and third line 4 entries, see keyword *elastic, type=anisotropic
      entry = 0
      for i in C_ave:
        entry = entry + 1
        ifile.write( i +'\t,')
        if entry == 8:
           ifile.write('\n')
           entry = 0
      ifile.write('\n')
    # finally write laminate and job information
    with open( material_jobData_filename, 'r') as material_jobData:
      for line in material_jobData:
        ifile.write( line )



def writeSaves( martensiteAmount, evaluationData, chem_drivForce, foundGrain ):
  ''' write data of Energy minimizing-configuration in the two files 'save_gain'
  containing grain specific data and 'save_model' containing model specific data.
  Also save the information of all other transformations in an increment as a reference.
  foundGrain = [0 - delta_allEner, 1 - GrainNr, 2 - GrainLaminate, 3 - GrainVol,
              4 - dragEner_spec, 5 - delta_totalStrain_spec,
              6 - stress_drivingForce, 7 - total_strainEner_cell ] '''
  #
  # write data from all runs of the actual increment
  with open('saves/allruns_'+str(martensiteAmount), 'w') as save_all:
    save_all.write('grainNr\tlaminateNr\tgrainVol\tdragEner_spec\t\t' + \
      'delta_totStrainEner_spec\tdelta_allEnergies\n')
    for dat in evaluationData:
      save_all.write( str(dat[1])+'\t'+str(dat[2])+'\t\t'+str(dat[3])+'\t'+ \
          str(dat[4])+'\t'+str(dat[5])+'\t\t'+str(dat[0])+'\n' )
  # write grain data:
  with open('saves/save_grain', 'a') as save_grain:
    save_grain.write(str(fg[1])+'\t'+str(fg[2])+'\t\t'+str(fg[3])+'\t'+str(fg[4])+ \
          '\t'+str(fg[5])+'\t\t'+str(fg[0])+'\t'+str(- hD + chem_drivForce )+'\n' )
  # gather and write model data:
  odbname = 'Outputfile_'+ str(martensiteAmount)+'_'+ str( fg[1] )+ \
        '_'+ str( fg[2] )+'.odb'
  md = automate.evaluate_odb( odbname, var = 0 ) # md ... modeldata
```

```
  md_6 = fg[3]*fg[5] # this is delta_total_strainEner
  # md = [0 - tot_strainEner, 1 - tot_aveSener, 2 - ivol_aust, 3 - ivol_mart,
  # 4 - aveSener_aust, 5 - aveSener_mart ]
  with open('saves/save_model', 'a') as save_model:
    save_model.write(str(md[0])+'\t'+str( md_6)+'\t'+str(md[1])+'\t'+str(md[2])+'\t'+ \
        str(md[3])+'\t'+str(md[4])+'\t'+str(md[5])+'\n')
```

E.3. automate.py

```
''' This module automates the calculation of all generated inputfiles as well as the
evaluation of the Outputdatabases. Also preselection parameters are specified.'''

import os
import time
import sys # to exit the subprocess after the abaqus standard job has finished
from multiprocessing import Process # manages subprocesses calculating the abaqus jobs
import psutil # library for retrieving information on running processes
# my modules
import mathutils


def preselect( totalGrainAmount, martensiteAmount ):
  ''' This function reduces the number of calculations carried out in a increment
  by preselecting more likely states known from a previous increment. The here
  defined parameters reduce the number of total calculations for this simulation
  by a factor of 6 while the results are the same. Note that especially, low
  fractions at early increments lower the number of total calculations because the
  number of not transformed grains and all possibilites are related multiplicatively.'''

  # here the fraction of all possible transformations is declared
  if martensiteAmount < int(totalGrainAmount*.7):
    calc_fraction = (1./7)
  if martensiteAmount>=int(totalGrainAmount*.7) and martensiteAmount<int(totalGrainAmount*.9):
    calc_fraction = (1./5)
  if martensiteAmount>=int(totalGrainAmount*.9) and martensiteAmount<int(totalGrainAmount*.94):
    calc_fraction = (1./2)
  if martensiteAmount >= int(totalGrainAmount*.95):
    calc_fraction = 1
  deltas = []
  # variant_preselection is taken from the last state in which
  # all variant permutations are known
  allstates = 'saves/allruns_' + str( martensiteAmount - 1 )
  with open(allstates,'r') as allstates:
    for index, line in enumerate( allstates ):
      line = line.rstrip('\n')
      if index == 0: continue # ignore the headerline
      else:
        # data = [ delta_ener, grain_nr, laminate_nr ]
        data = [ float(line.split()[5]), line.split()[0], line.split()[1] ]
        deltas.append( data )
        del data
  deltas.sort() # sort ascending
  deltas.reverse() # reverse to get descending sort
  amount = int( len(deltas)* calc_fraction )
  deltas = deltas[ 0 : amount ]
  preselection = []
  for i in deltas:
    preselection.append( [ int(i[1]), int(i[2]) ] ) # [grainNr, laminate]
  return preselection
```

```
52 def submitjobs( austeniteGrains, martensiteAmount, laminateVariants, preselection, \
53                 selected_variants, timeout ):
54   ''' handles automatic submission of all inputfiles, created in an increment '''
55   #
56   wait_array = []
57   jobNr = 0
58   #
59   for austeniteGrain in austeniteGrains:
60     for laminate in laminateVariants:
61       if ( [austeniteGrain[0], laminate] in selected_variants) or preselection == False:
62         #
63         jobNr = jobNr + 1
64         inputname = 'Inputfile_'+ str(martensiteAmount)+'_'+ \
65         str(austeniteGrain[0])+ '_'+ str(laminate) +'.inp'
66         outputname = 'Outputfile_'+ str(martensiteAmount)+ '_'+ \
67         str( austeniteGrain[0] )+'_'+ str(laminate)
68         #
69         pid = os.fork() # start (=fork) subprocesses
70         if pid == 0: # 0 is the child process
71           # in the child process invoke the standard solver
72               os.system( '/opt/abaqus/Commands/abq6123 job=' + outputname + \
73             ' interactive cpus=2 scratch=/dev/shm input=' + \
74             inputname +' mp_mode=threads standard_parallel=all')
75           # /dev/shm tmpfs directory
76           sys.exit() # close subprocess after the job has finished
77         else: # in the parent process the child process are managed
78           process = psutil.Process( pid )
79           wait_array.append( process)
80         #
81         if jobNr % 6 == 0 or (preselection == True and jobNr == len(selected_variants) ):
82           for iprocess in wait_array:
83             try:
84               iprocess.wait( timeout )
85               # timeout is the time in seconds the script waits for completion of the job
86             except psutil.TimeoutExpired:
87               print 'process running after timeout, killing proces...'
88               os.system('killall -9 standard.exe')
89               time.sleep(30)
90           wait_array = [] # empty wait array for new processes
91   del inputname, outputname, wait_array, jobNr


94 def findMinimumEnergy( austeniteGrains, martensiteAmount, laminateVariants, preselection, \
95                       selected_variants = [], total_strainEner_cell_before = 0, \
96                       chemical_drivingForce = 0 ):
97   ''' Reads totalstrainergy from outputfiles and evaluates the transformation that minimizes
98       the total strain energy density '''
99   #
100  evaluationData = [] # Define list for calculation results
101  #
102  for austeniteGrain in austeniteGrains:
103    for laminate in laminateVariants:
104      if ( [austeniteGrain[0], laminate] in selected_variants) or preselection == False:
105        #
106        odbname = 'Outputfile_'+ str(martensiteAmount)+'_'+ str( austeniteGrain[0] )+ \
107               '_'+ str(laminate)+'.odb'
108        #
109        # if the calculation was terminated and a .lck file exist ignore that .odb
110        if os.path.isfile( odbname[0:len(odbname)-3] +'lck' ): continue
111        # read the totalstrainenergy of the whole model via the odb file
112        # alternatively it could be extracted from the .dat file
```

```
113         else: total_strainEner_cell = evaluate_odb.(odbname, var=1)
114         #
115         # Calculate difference of free energy density to previous increment:
116         # first calculate specific strain energy of transformed grain
117         delta_totalStrain = total_strainEner_cell - total_strainEner_cell_before
118         delta_totalStrain_spec = delta_totalStrain / austeniteGrain[1]
119         # next calculate specific interface energy barrier of transformed grain
120         dragEner_spec = forces.calc_draggingForces( austeniteGrain[1] )
121         #
122         # In the first run the chemical_drivingForce is determined as the sum
123         # of dragging energies, thus it a negative value
124         delta_G = chemical_drivingForce - ( dragEner_spec + delta_totalStrain_spec )
125         # CAUTION! : line continuation with - \ - gives +! 1--1 = 2!
126         #
127         evaluationData.append( [delta_G, austeniteGrain[0], laminate, austeniteGrain[1], \
128               dragEner_spec, delta_totalStrain_spec, total_strainEner_cell ] )
129         #
130         del total_strainEner_cell, delta_totalStrain, delta_totalStrain_spec, \
131           dragEner_spec, delta_G
132   return evaluationData


135 def evaluate_odb(odbname, var = 0):
136   ''' this function has three different return values (var = 0, 1, 2)
137   per default (0) weighted strain energy densities are returned. var = 1: only the total
138   strain energy of the model is returned. var = 2: The transformation criterion for
139   the LTC is returned based on the specific IE energy barrier and the double dot product
140   of the averaged strain  tensor in a grain and its possible eigenstrains'''
141   # create odb singular object
142   odb = openOdb(path=odbname)
143   # go through the object model
144   trans_step = odb.steps['Transformation']
145   last_frame = trans_step.frames[-1] # [-1] gives last frame

147   if var == 1:
148     histreg = trans_step.historyRegions['Assembly Assembly-1']
149     # Assembly Assembly-1 is the default repository key that is generated
150     all_total_energies = histreg.historyOutputs['ALLIE'].data
151     # .data is the key in the dictionary for the Allenergies (total energies) array.
152     odb.close()
153     return all_total_energies[1][1] # [1][1] is the totalstrainenergy

155   # --- integration point variables ---
156   stresses = last_frame.fieldOutputs['S'] # stress tensor components of integration points
157   seners = last_frame.fieldOutputs['SENER'] # strain energy densities of integration points
158   ivols = last_frame.fieldOutputs['IVOL'] # integration point volumes
159   '''# --- alternatively wole element variables can be read ---
160         last_frame.fieldOutputs['ESEDEN'] # equivalent to SENER for whole elements
161   last_frame.fieldOutputs['ELSE'] # strain energy for all whole Elements
162   last_frame.fieldOutputs['EVOL'] # equivalent to IVOL for whole elements '''
163   #
164   # initialize variables to be weighted from the integration point level
165   total_strainEner = total_aveSener = ivol_total = ivol_aust = ivol_mart = 0
166   tot_strainEner_aust = tot_strainEner_mart = 0
167   #
168   for igrain in odb.sections.values():
169   # or: for i in odb.rootAssembly.elementSets.keys() -> i = set_name from inputfile
170     #
171     material = igrain.material
172     set_name = igrain.name.lstrip('Section-')
173     #
174     # 'PART-1-1' is the default name of the first created part if none is specified
```

```
175   grain = odb.rootAssembly.instances['PART-1-1'].elementSets[set_name]
176   # integration point variable subsets
177   set_stresses = stresses.getSubset(region = grain, position=INTEGRATION_POINT)
178   set_ivol = ivols.getSubset(region = grain, position=INTEGRATION_POINT)
179   set_sener = seners.getSubset(region = grain, position=INTEGRATION_POINT)
180   # for whole element variables the same can be done without the parameter position=...
181   #
182   # evaluate the elementset related to the section
183   grain_sener_sum = grain_ges_ivol = 0
184   for i in range(len(set_stresses.values)): # number of all integration points
185     # in the set SENER must be weighted with the element volume since
186     # not elements are of the same size. Same as ELSE
187         grain_sener_sum=grain_sener_sum+set_sener.values[i].data*set_ivol.values[i].data
188     grain_ges_ivol = grain_ges_ivol + set_ivol.values[i].data
189   grain_ave_sener = grain_sener_sum / grain_ges_ivol
190   total_strainEner = total_strainEner + grain_sener_sum
191   # the volume of the matrix must not be considered for the random RVE cell !
192   if 'TRANSIG' in set_name: ivol_total = ivol_total + grain_ges_ivol
193   #
194   # additionally calculate the strain energy developement in each phase respectively
195   if material == 'AUSTENITE':
196     ivol_aust = ivol_aust + grain_ges_ivol
197     tot_strainEner_aust = tot_strainEner_aust + grain_ave_sener * grain_ges_ivol
198   #
199   if 'LAMINATE' in material:
200     ivol_mart = ivol_mart + grain_ges_ivol
201     tot_strainEner_mart = tot_strainEner_mart + grain_ave_sener * grain_ges_ivol

203   if var == 2 and material == 'AUSTENITE':
204     # initialize transformation strains
205     transforming_strains = eigenstrains()
206     # Define average "effective" stress tensor components
207     sig_eff_11 = sig_eff_22 = sig_eff_33 = sig_eff_12 = sig_eff_13 = sig_eff_23 = 0
208     #
209     for i in range(len(set_stresses.values)):
210       grain_ges_ivol = grain_ges_ivol + set_ivol.values[i].data
211       #
212       sig_eff_11 = sig_eff_11 + set_stresses.values[i].data[0]*set_ivol.values[i].data
213           sig_eff_22 = sig_eff_22 + set_stresses.values[i].data[1]*set_ivol.values[i].data
214           sig_eff_33 = sig_eff_33 + set_stresses.values[i].data[2]*set_ivol.values[i].data
215           sig_eff_12 = sig_eff_12 + set_stresses.values[i].data[3]*set_ivol.values[i].data
216           sig_eff_13 = sig_eff_13 + set_stresses.values[i].data[4]*set_ivol.values[i].data
217           sig_eff_23 = sig_eff_23 + set_stresses.values[i].data[5]*set_ivol.values[i].data
218           #
219     sig_eff_11 = sig_eff_11 / grain_ges_ivol
220     sig_eff_22 = sig_eff_22 / grain_ges_ivol
221     sig_eff_33 = sig_eff_33 / grain_ges_ivol
222     sig_eff_12 = sig_eff_12 / grain_ges_ivol
223     sig_eff_13 = sig_eff_13 / grain_ges_ivol
224     sig_eff_23 = sig_eff_23 / grain_ges_ivol
225     #
226     sigma = [sig_eff_11, sig_eff_22, sig_eff_33, sig_eff_12, sig_eff_13, sig_eff_23]
227     sigma = mathutils.fillmatrix(sigma)
228     #
229     spec_grain_drag = forces.calc_draggingForces( grain_ges_ivol )
230     for laminate in range(1, 6+1): # = [1,2,3,4,5,6]
231       spec_driving_stress=mathutils.doubledotproduct(transforming_strains[laminate-1],sigma)
232       check_term = spec_driving_stress - spec_grain_drag
233       if check_term > driving_force
234         driving_force = check_term
235         found_grain = [set_name, laminate, driving_force]
236 odb.close()
```

```
237   #
238   # calculate total model values
239   total_ivol = ivol_aust + ivol_mart
240   # note that for the random RVE this is only the graincluster without the matrix
241   total_aveSener = total_strainEner / ivol_total
242   total_aveSener_mart = total_strainEner_mart / ivol_mart
243   if ivol_aust != 0: total_aveSener_aust = total_strainEner_aust / ivol_aust

245   if var == 2:
246     return found_grain, total_strainEner, total_aveSener, ivol_austenite, ivol_transformed, \
247           tot_aveSener_aust, tot_aveSener_mart

249   return total_strainEner, total_aveSener, ivol_austenite, ivol_transformed, \
250         tot_aveSener_aust, tot_aveSener_mart
```

E.4. material.py

```
1  ''' It also holds the parameters for the semianalytical calculation of
2  all IEs constituting a part of the energy barrier in the transformation '''

4  import numpy as np # used for a faster function of tensor rotation using outer products
5  import math
6  #my modules
7  import mathutils

9  def eigenstrains():
10 ''' initializes the matrices representing the transformation strains'''
11   stress_driving_forces = []
12   # transformation strain components
13   [ v, u, w ] = [ 0.0381, 0.0234 0.1186 ]
14   # e1 corresponds to laminate 1, e2 to laminate 2 etc.
15   e1 = [ u, -v, u, 0,-w, 0 ]
16   e2 = [ u, u, -v, w, 0, 0 ]
17   e3 = [ u, -v, u, 0, w, 0 ]
18   e4 = [ u, u, -v,-w, 0, 0 ]
19   e5 = [ v, u, u, 0, 0, -w ]
20   e6 = [ -v, u, u, 0, 0, w ]
21   e = [ e1, e2, e3, e4, e5, e6]
22   transforming_strains = []
23   for i in e:
24     transforming_strains.append( mathutils.fillmatrix( i ) )
25   return transforming_strains


28 def calc_draggingForces( volume_grain ):
29   ''' Given the grainvolume of a certain grain in [nm^3] the specific dragging forces for
30   the grain are calculated. Therefore the grain is assumed to be a sphere with equal volume
31   as the grain. '''
32   #
33   # ----- < P A R A M E T E R S > -----
34   sigma_twin = 0.014E-9 # [nJ/nm^2] specific twin interfaceenergy
35   poissons_ratio_austenite = 0.4
36   alpha_twin = 0.856 * (1+poissons_ratio_austenite) / (8*math.pi)
37   shear_distortion = 0.1677 # 2*math.sqrt( e12**2 + e23**2 )
38   youngs_modulus_austenite = 72.E-9
39   shear_modulus_austenite = youngs_modulus_austenite / (2*(1+poissons_ratio_austenite))
40   inclusion_energy = 2.42E-10 # 1.177E-10
41   fit_C = inclusion_energy / (alpha_twin*2*1.5*shear_distortion**2 * shear_modulus_austenite)
42   # 1.5nm = d, 2d = D, the thickness of the 1nm model is not written explicitly
```

```
  Fc = 5.8E-12 # [nJ/nm^3] work of friction, dissipated energy
  delta_surfaceEnergy = 0.1E-9 # [nJ/nm^2]; values from 0.1 ... 0.4 are reasonable
  #
  # At first the equal sphere's radius and surface is calculated
  diameter, surf = equal_lengths(volume_grain)
  # next the optimal twin width minimizing the dragging force is calculated
  d_opt = math.sqrt( sigma_twin / (12 * fit_C * alpha_twin * shear_modulus_austenite * \
                  shear_distortion**2) ) * math.sqrt(diameter)
  #
  # The dragging forces follow as:
  total_interfaceEnergy = ( diameter / (6*d_opt) ) * sigma_twin
  twin_surfaceEnergy = fit_C*alpha_twin*shear_modulus_austenite*shear_distortion**2 *2 *d_opt
  dragging_forces = (total_interfaceEnergy + delta_surfaceEnergy + twin_surfaceEnergy) \
                  * (surf / volume_grain) + Fc
  return dragging_forces


def equal_lengths(volume):
  ''' given the volume of a grain calculates the diameter
  and surface of a sphere with equal volume '''
  diameter = ( (6*volume) / math.pi ) ** (1./3.)
  surf = diameter**2 * math.pi
  return diameter, surf


def selfconsistent_matrix( oris, graindata, Vinner ):
  '''This function averages anisotropic elastic constants ( refering to local coordinate
  systems respectively) to global isotropic elastic constants considering phase fractions.
  The nearly isotropic elastic constants are used as the matrix material property. In
  micromechanics this is commonly called "self consistence scheme" '''
  #
  # define isotropic elastic constants for austenite
  E_aust = 70e-9
  poissons_ratio_aust = 0.4
  prefactor_austenite = E_aust / (( 1 + poissons_ratio_aust)*( 1 - 2*poissons_ratio_aust))
  # Assemble isotropic elastic constants of austenite as a fourth order tensor
  # note that the the following lines hold the 21 independent entries of the elastic tensor
  # respectively. The order is the same as in the abaqus inputfile.
  # first line in inputfile
  A1111 = prefactor_austenite * (1 - poissons_ratio_aust)
  A1122 = A2211 = prefactor_austenite * poissons_ratio_aust
  A2222 = prefactor_austenite * (1 - poissons_ratio_aust)
  A1133 = A3311 = prefactor_austenite * poissons_ratio_aust
  A2233 = A3322 = prefactor_austenite * poissons_ratio_aust
  A3333 = prefactor_austenite * (1 - poissons_ratio_aust)
  A1112 = A1211 = A1121 = A2111 = 0.
  A2212 = A1222 = A2221 = A2122 = 0.
  # second line in inputfile
  A3312 = A1233 = A3321 = A2133 = 0.
  A1212 = A2112 = A1221 = A2121 = prefactor_austenite * ((1 - 2*poissons_ratio_aust) / 2)
  A1113 = A1311 = A1131 = A3111 = 0.
  A2213 = A1322 = A2231 = A3122 = 0.
  A3313 = A1333 = A3331 = A3133 = 0.
  A1213 = A1312 = A2113 = A1231 = A3112 = A1321 = A2131 = A3121 = 0.
  A1313 = A3113 = A1331 = A3131 = prefactor_austenite * ((1 - 2*poissons_ratio_aust) / 2)
  A1123 = A2311 = A1132 = A3211 = 0.
  # thirA line in inputfile
  A2223 = A2322 = A2232 = A3222 = 0.
  A3323 = A2333 = A3332 = A3233 = 0.
  A1223 = A2312 = A2123 = A1232 = A3212 = A2321 = A2132 = A3221 = 0.
  A1323 = A2313 = A3123 = A1332 = A3213 = A2331 = A3132 = A3231 = 0.
  A2323 = A3223 = A2332 = A3232 = prefactor_austenite * ((1 - 2*poissons_ratio_aust) / 2)
```

```
105  # anisotropic elastic constants of martensite given in the basis of the tetragonal unit cell
106  M1111 = 2.54e-07
107  M1122 = M2211 = 1.04e-07
108  M2222 = 1.8e-07
109  M1133 = M3311 = 1.36e-07
110  M2233 = M3322 = 1.51e-07
111  M3333 = 2.48e-07
112  M1112 = M1211 = M1121 = M2111 = 0.
113  M2212 = M1222 = M2221 = M2122 = 0.
114  M3312 = M1233 = M3321 = M2133 = 0.
115  M1212 = M2112 = M1221 = M2121 = 9.1e-8
116  M1113 = M1311 = M1131 = M3111 = 0.
117  M2213 = M1322 = M2231 = M3122 = 0.
118  M3313 = M1333 = M3331 = M3133 = 0.
119  M1213 = M1312 = M2113 = M1231 = M3112 = M1321 = M2131 = M3121 = -3.e-9
120  M1313 = M3113 = M1331 = M3131 = 9.3e-08
121  M1123 = M2311 = M1132 = M3211 = 2.1e-08
122  M2223 = M2322 = M2232 = M3222 = 0.
123  M3323 = M2333 = M3332 = M3233 = -6.e-09
124  M1223 = M2312 = M2123 = M1232 = M3212 = M2321 = M2132 = M3221 = 0.
125  M1323 = M2313 = M3123 = M1332 = M3213 = M2331 = M3132 = M3231 = 0.
126  M2323 = M3223 = M2332 = M3232 = 5.e-09
127  #
128  Ca =[ [ [ [A1111, A1112, A1113], [A1121, A1122, A1123], [A1131, A1132, A1133] ],
129          [ [A1211, A1212, A1213], [A1221, A1222, A1223], [A1231, A1232, A1233] ],
130          [ [A1311, A1312, A1313], [A1321, A1322, A1323], [A1331, A1332, A1333] ] ],
131        [ [ [A2111, A2112, A2113], [A2121, A2122, A2123], [A2131, A2132, A2133] ],
132          [ [A2211, A2212, A2213], [A2221, A2222, A2223], [A2231, A2232, A2233] ],
133          [ [A2311, A2312, A2313], [A2321, A2322, A2323], [A2331, A2332, A2333] ] ],
134        [ [ [A3111, A3112, A3113], [A3121, A3122, A3123], [A3131, A3132, A3133] ],
135          [ [A3211, A3212, A3213], [A3221, A3222, A3223], [A3231, A3232, A3233] ],
136          [ [A3311, A3312, A3313], [A3321, A3322, A3323], [A3331, A3332, A3333] ] ] ]
137  #
138  Cm =[ [ [ [M1111, M1112, M1113], [M1121, M1122, M1123], [M1131, M1132, M1133] ],
139          [ [M1211, M1212, M1213], [M1221, M1222, M1223], [M1231, M1232, M1233] ],
140          [ [M1311, M1312, M1313], [M1321, M1322, M1323], [M1331, M1332, M1333] ] ],
141        [ [ [M2111, M2112, M2113], [M2121, M2122, M2123], [M2131, M2132, M2133] ],
142          [ [M2211, M2212, M2213], [M2221, M2222, M2223], [M2231, M2232, M2233] ],
143          [ [M2311, M2312, M2313], [M2321, M2322, M2323], [M2331, M2332, M2333] ] ],
144        [ [ [M3111, M3112, M3113], [M3121, M3122, M3123], [M3131, M3132, M3133] ],
145          [ [M3211, M3212, M3213], [M3221, M3222, M3223], [M3231, M3232, M3233] ],
146          [ [M3311, M3312, M3313], [M3321, M3322, M3323], [M3331, M3332, M3333] ] ] ]
147  # create numpy arrays
148  Ca = np.array(Ca)
149  Cm = np.array(Cm)
150  C_ave = np.zeros((3,3,3,3))
151  for i in range(len(graindata)):
152    # get rotationmatrix between local and global coordinate system.
153    rot_euler = np.array( mathutils.calc_rotmatrix_euler( oris[i][0], oris[i][1] ) )
154    # distinguish between austenite and martensite
155    if graindata[i][2] == 'AUSTENITE': C = Ca
156    else: C = Cm
157    Vi = graindata[i][1]
158    # calculate C_averaged = sum_x[ (Vx/Vinner) * ( Rim Rjn Rkp Rlq Cmnpq ) ]
159    # where Cmnpq can be C_a or C_m
160    C_ave = np.add( C_ave, np.multiply( (Vi/Vinner), rotateElasticTensor(C,rot_euler) ) )
161  C_selfconsistent_voigt = voigt_notation( C_ave )
162  # convert float entries to strings with five decimals in order to write to inputfile
163  for i in range( len(C_selfconsistent_voigt) ):
164    C_selfconsistent_voigt[i] = '{0:.5e}'.format(float( C_selfconsistent_voigt[i]) )
165    # positional argument 0 in python 2.x required.
166  return C_selfconsistent_voigt
```

E.5. mathutils.py

```
''' This module contains all mathematical operations needed for the simulation '''

import math
import izip
import numpy as np # installed along with abaqus. Available after invoking abaqus python

# -----< used matrix operations >---------------------------------------------------------#

def doubledot_product(A, B):
  '''returns the doubledot product of two 3x3 matrices'''
  result = 0
  for i in range(3):
    for j in range(3):
      result = result + A[i][j]*B[i][j]
  return result

def fillMatrix(A):
  ''' creates a symmetric 3x3 matrix from its elements
  A = [11,22,33,12,13,23] '''
  result = [ [A[0],A[3],A[4]],[A[3],A[1],A[5]],[A[4],A[5],A[2]] ]
  return result

# -----< used vector operations >---------------------------------------------------------#

def vecDotProduct(vec1,vec2):
  ''' calculates the dot or scalar product of two arbitrary vectors'''
  return sum(x1 * x2 for x1, x2 in izip(vec1, vec2))

def vecNorm(vec):
  ''' calculates the Euclidean norm of a vector '''
  return vecDotProduct(vec,vec)**0.5

def angle_between_vectors(vec1,vec2):
  '''calculates the angle between two vectors in radians'''
  return math.acos( vecDotProduct(vec1,vec2) / ( vecNorm(vec1)*vecNorm(vec2) ) )

def vecCrossProduct(vec1, vec2):
  ''' calculates the cross product between two vectors '''
  return( [ vec1[1] * vec2[2] - vec1[2] * vec2[1],
            vec1[2] * vec2[0] - vec1[0] * vec2[2],
            vec1[0] * vec2[1] - vec1[1] * vec2[0])

# -----[ calculation of rotation matrix relating two Cartesion coordinate systems and ]---#
# [ rotations of fourth order tensors from one coordinate system to the other ]

def voigt_notation( C )
  ''' this function takes the elastic fourth order tensor and returns its Voigt notation,
      which is a 6 x 6 matrix '''
  return C_voigt = [ C[0][0][0][0],C[0][0][1][1],C[1][1][1][1],
                     C[0][0][2][2],C[1][1][2][2],C[2][2][2][2],
                     C[0][0][0][1],C[1][1][0][1],C[2][2][0][1],
                     C[0][1][0][1],C[0][0][0][2],C[1][1][0][2],
                     C[2][2][0][2],C[0][1][0][2],C[0][2][0][2],
                     C[0][0][1][2],C[1][1][1][2],C[2][2][1][2],
                     C[0][1][1][2],C[0][2][1][2],C[1][2][1][2] ]
```

```
59 def calc_rotmatrix_euler(a,b):
60   '''Calculates the rotation matrix R that takes the Cartesion coordinate system
61   defined by the two normal unit vectors a and b to the reference system given by
62   x[1,0,0] y[0,1,0] z[0,0,1]. R is defined using three Eulerian angles whereby
63   the x-convention (z,y',z'') is used. '''
64   #
65   # the third vector of the rotated system is calculated
66   c = vecCrossProduct(a,b)
67   # the Eulerian angles are calculated
68   alpha, beta, gamma = eulerian_angles(a,b,c)
69   # the Rotationsmatrix entrys are given as:
70   r11 = - math.sin(alpha)*math.sin(gamma) + math.cos(alpha)*math.cos(beta)*math.cos(gamma)
71   r12 = math.cos(alpha)*math.sin(gamma) + math.sin(alpha)*math.cos(beta)*math.cos(gamma)
72   r13 = - math.sin(beta)*math.cos(gamma)
73   r21 = - math.sin(alpha)*math.cos(gamma) - math.cos(alpha)*math.cos(beta)*math.sin(gamma)
74   r22 = math.cos(alpha)*math.cos(gamma) - math.sin(alpha)*math.cos(beta)*math.sin(gamma)
75   r23 = math.sin(beta)*math.sin(gamma)
76   r31 = math.cos(alpha)*math.sin(beta)
77   r32 = math.sin(alpha)*math.sin(beta)
78   r33 = math.cos(beta)
79   return ((r11, r12, r13),(r21, r22, r23),(r31, r32, r33))

81 def eulerian_angles(a,b,c):
82   ''' calculates the eulerangles between two Cartesian coordinate systems with
83   the same origin. The 3 Eulerian angles are dependent from each other, which means
84   the order of plane rotations carried out to transform the coordinate system is
85   definite. The angle beta is simply the angle between the z axes of both coordinate
86   systems (Z and c here). The angle alpha is the angle between the X axis of the
87   reference coordinate system and the projection of c into the X,Y plane. Finally,
88   gamma is the angle between the b - axis and Y'
89   '''
90   beta = angle_between_vectors( (0.,0.,1.),c )
91   alpha = angle_between_vectors( (1.,0.,0.), (c[0],c[1],0) )
92   # for gamma Y has to be rotated first
93   Y_rotated = [0,0,0]
94   Y_rotated[0] = - math.sin(alpha) * 1
95   Y_rotated[1] = math.cos(alpha) * 1
96   gamma = angle_between_vectors(Y_rotated, b)
97   return alpha, beta, gamma

99 def rotate_indicial( C, R ):
100   ''' calculates the rotation of a fourth order tensor C by the rotation defined by R,
101       where both are given as standard python lists '''
102   for i in range(3):
103     for j in range(3):
104       for k in range(3):
105         for l in range(3):
106           for m in range(3):
107             for n in range(3):
108               for o in range(3):
109                 for p in range(3):
110                   Crot[i][j][k][l]=R[i][m]*R[j][n]*R[k][o]*R[l][p]*C[m][n][o][p]+Crot[i][j][k][l]
111   return Crot

113 def rotateElasticTensor(C, R):
114     ''' rotates a forth order tensor C by a rotation given by the matrix R.
115   see: numpy reference-linear-algebra p653 '''
116     RR = np.outer(R, R)
117     RRRR = np.outer(RR, RR).reshape(4 * R.shape)
118     axes = ((0, 2, 4, 6), (0, 1, 2, 3))
119     return np.tensordot(RRRR, C, axes)
```

List of Figures

List of Tables

Bibliography

[1] M. Cohen, G.B. Olson, and P.C. Clapp, editors. *International Summer Course on Martensitic Transformations*, volume 79, Leuven: Dep. Met. & Mat. Eng., 1979.

[2] L. Delaey, K. Mukherjee, and M. Chandrasekaran, editors. *International Summer Course on Martensitic Transformations*, Leuven: Dep. Met. & Mat. Eng., 1982b.

[3] A.L. Roitburd and G.V. Kurdjumov. The nature of martensitic transformations. *Materials Science and Engineering*, 39:141–167, 1979.

[4] J.A. Klostermann. The concept of the habit plane and the phenomenological theories of the martensite transformation. *Journal of the less-common metals*, 28:75–94, 1972.

[5] C. Zener. *Elasticity and anelasticity of Metals*. Universitiy of Chicago Press, Chicago, 1948.

[6] E. Hornbogen and H. Warlimont. *Metalle*, volume 5. Springer-Verlag Berlin Heidelberg, 2006.

[7] L.D.Landau and E.M.Lifschitz. *Course of theoretical Physics - Statistical Physics, Part I.* Nauka, Moscow, 1976.

[8] G. Kostorz. *Phase transformations in materials*. University Press, D-69469 Weinheim Germany, 1968.

[9] K. Bhattacharya, S. Conti, and G. Zanzotto. Crystal symmetry and the reversibility of martensitic transformations. *Nature*, 428:55–59, 2004.

[10] K. Bhattacharya. Comparison of the geometrically nonlinear and linear theories of martensitic transformation. *Continuum Mechanics and Thermodynamics*, 5(3):205–242, 1993.

[11] J.S. Bowles and J.K. Mackenzie. The crystallographie of martensite transformations I. *Acta Met.*, 2:129–137, 1954.

[12] J.S. Bowles and J.K. Mackenzie. The crystallographie of martensite transformations II. *Acta Met.*, 2:137–147, 1954.

[13] M.S. Wechsler, D.S. Liebermann, and T.A. Read. On the theory of the formation of martensite. *Trans AIME J. Metals*, 197:1503, 1953.

[14] J.L. Ericksen. On the Chauchy - Born rule. *Math. and Mech. of Solids*, 13:199, 2008.

[15] J.L. Ericksen. On correlating two theories of twinning. *Arch. Rat. Mech. Anal.*, 153:261–289, 2000.

[16] M. Pitteri and G. Zanzotto. *Continuum Models for Phase Transitions and Twinning in Crystals*. Chapman and Hall, London, 1998.

[17] G. Zanzotto. On the material symmetry group of elastic crystals and the born rule. *Arch. Rational Mech. Anal*, 121:1–36, 1992.

[18] J.M. Ball and R.D. James. Fine phase mixtures as minimizers of energy. *Arch. Rational Mech. Anal*, 100:15–52, 1989.

[19] J.M. Ball. Mathematical models of martensitic microstructure. *Mat. Sci. Eng.*, A 378:61–69, 2004.

[20] K. Otsuka and X. Ren. Physical metallurgy of Ti-Ni-based shape memory alloys. *Prog Mater Sci*, 50:511–678, 2005.

[21] K. Bhattacharya. *Microstructure of martensite*. University Press, Oxford, 2003.

[22] R.D.James and K.F. Hane. Martensitic transformations and shape memory materials. *Acta Mat.*, 48:197–222, 2000.

[23] J. Perkins. *Shape memory effects in alloys*. Plenum press London, London, 1975.

[24] J.W. Christian. *The Theory of Phase Transformations in Metals and Alloys*. Pergamon Press, Oxford, 1965.

[25] E. Schmid and W. Boas. *Plasticity of crystals: with special reference to metals*. Chapman & Hall, 1968.

[26] D.C. Lagoudas. *Shape Memory Alloys Modeling and Engineering Applications*. Springer press, New York, 2008.

[27] T. Massalski, H. Okamoto, P. Subramanian, and L. Kacprzac. *Binary Alloy Phase Diagrams Vol.3*. ASM International, Ohio, 1990.

[28] K. Otsuka and X. Ren. Martensitic transformations in nonferrous shape memory alloys. *Mat. Sci. Eng. A*, A273-275:89, 1999.

[29] G. Bastin and G. Rieck. Occurance of growth of the various intermetallic compounds. *Metall. Trans.*, 5:1817, 1974.

[30] W. Tang and B. Sundman. New modelling of the b2 phase and its associated martensitic transformation in the NiTi-system. *Acta Mater.*, 47:3457, 1999.

[31] *Thermomechanische Behandlung von NiTi*, number 462 in 5. VDI-Verlag Düsseldorf, 1997.

[32] K. Gall and H. Sehitoglo. Tension-compression asymmetry of the stress-strain response in aged single crystal and polycrystalline NiTi. *Acta Mater.*, 47:1203–1217, 1999.

[33] S. Miyazaki and K. Otsuka. Deformation and transformation behavior associated with the r-phase in ti-ni alloys. *Metallurgical Transactions*, 17A:53–63, 1986.

[34] H. Liu and L. Mishnaevsky. Martensitic transformations in nanostructured nitinol: Finite element modeling of grain size and distribution effects. *Comp. Mat. Sci.*, 76:27–36, 2013.

[35] M. Peterlechner, T. Waitz, C. Gammer, and T. Antretter. Martensitic phase transformations of nanocrystalline NiTi shape memory alloys processed by repeated cold rolling. *Int. J. Mat. Res.*, 6:102, 2011.

[36] C. Carlton and P.J. Ferreira, editors. *What is Behind the Inverse Hall-Petch Behavior in Nanocrystalline Materials*, volume 976. Materials Research Society, 2007.

[37] H. Gleiter. On the structure of grain boundaries in metals. *Materials Science and Engineering*, 52(2):91–131, 1982.

[38] M. Zehetbauer, R. Grössinger, H. Krenn, M. Krystian, R. Pippan, P. Rogl, T. Waitz, and R. Würschum. Bulk nanostructured functional materials by severe plastic deformation. *Advanced engineering materials*, 12(8), 2010.

[39] T. Waitz. The self-accommodeted morphology of martensite in nanocrystalline NiTi shape memory alloys. *Acta Mater.*, 53:2273–2283, 2005.

[40] T. Waitz, V. Kazykhanov, and H.P. Karnthaler. Martensitic phase transformations in nanocrystalline NiTi studied by tem. *Acta Mater.*, 52:137–147, 2004.

[41] W. Pranger. Numerische analyse der martensit morphologie in nanokristallinen NiTi legierungen. Master's thesis, University of Leoben, Institute for Mechanik, 2011.

[42] N.B. Morgan. Medical shape memory alloy applications - the market and its products. *Material Science and Engineering*, 378:16–23, 2004.

[43] G. Mase. *Schaum's Outline of Continuum Mechanics*. Schaum's Outline Series. McGraw-Hill Education, 1970.

[44] Y. Kudoh, M. Tokonami, S. Miyazaki, and K. Otsuka. Crystal structure of the martensite in Ti-49.2at%Ni alloy analyzed by the single crystal x-ray diffraction method. *Acta Metall.*, 33:2049–2056, 1985.

[45] R. Golesorktabar. Ab initio calculation of elastic properties; general implementation and specific application to the shape-memory material NiTi. Master's thesis, University of Leoben, Chair of Atomistic Modelling and Design of Materials, 2013.

[46] S. Miyazaki, K. Kimura, K. Otsuka, and Y. Suzuki. The habit plane and transformation strains associated with the martensitic transformation in tini single crystals. *Scripta Metall*, 18:883, 1984.

[47] M. Pitteri and G. Zanzotto. Generic and non-generic cubic-to-monoclinic transitions and their twins. *Acta mater.*, 46:225–237, 1996.

[48] K. Bhattacharya. Wedge-like microstructure in martensites. *Acta Metall.*, 39:2431–2444, 1991.

[49] K.F. Hane and T.W. Shield. Microstructure in the cubic to monoclinic transition in titanium-nickel shape memory alloys. *Acta Mater.*, 47:2603–2617, 1999.

[50] A. Forclaz. *Variational methods in material science*. PhD thesis, University of Oxford, 2002.

[51] B. Dacorogna. *Direct Methods in the Calculus of Variations*. Springer, 2007.

[52] A.G. Khachaturyan. *Theory of Structural Transformations in Solids*. Dover Publications Inc., Mineola, New York, 1983.

[53] X. Chen, V. Srivastava, V. Dabade, and R.D. James. Study of the cofactor conditions: Conditions of supercompatibility between phases. *J. Mech. Phys. Solids*, 61:2566–2587, 2013.

[54] T. Waitz, T. Antretter, F.D. Fischer, N.K. Simha, and H.P. Karnthaler. Size effects on the martensitic phase transformation of NiTi nanograins. *J. Mech. Phys. Solids*, 55:419–444, 2007.

[55] R.Quey, P.R.Dawson, and F.Barbe. Large-scale 3d random polycrystals for the finite element method: Generation, meshing and remeshing. *Comput.Methods Appl.Mech.Engrg.*, 200:1729–1745, 2011.

[56] C. Geuzaine and J.F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *Int. Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

[57] R. Quey. *Documentation for Neper 2.0.1 A software package for polycrystal generation and meshing*, 2014. `http://neper.sourceforge.net/`.

[58] Python Software Foundation. Python 2 tutorial. `https://docs.python.org/2/tutorial/`. Accessed: 2014-04-05.

[59] Python Software Foundation. Python 3 tutorial. `https://docs.python.org/3/tutorial/`. Accessed: 2014-04-05.

[60] simulia. *Abaqus Documentation*, 2014. `http://www.3ds.com/products-services/simulia/portfolio/abaqus/overview/`.

[61] Y. Liu and P.G. McCormick. Thermodynamic analysis of the martensitic transformation in NiTi. effect of heat treatment on transformation behaviour. *Acta. Matell. Mater.*, 42:2406–2410, 1994.

[62] H.P. Karnthaler T. Waitz. Size-dependent martensitic transformation path causing atomic-scale twinning of nanocrystalline shape memory alloys. *Europhys. Lett.*, 71:98–103, 2005.

[63] M.X. Wagner and W. Windl. Lattice stability, elastic constants and macroscopic moduli of NiTi martensites from first principles. *Acta Materialia*, 56:59–60, 2008.

[64] N. Hatcher, O.Y. Kontsevoi, and A.J. Freeman. Role of elastic and shear stabilities in the martensitic transformation path of NiTi. *Phys. Rev.*, 80:59–60, 2009.

[65] M. Fukuhara, M. Yagi, and A. Matsuo. Temperature dependence of elastic parameters and internal frictions for NiTi alloy. *Phys. Rev. B*, 65:210–224, 2002.

[66] Q.S. Mei, L. Zhang, K. Tsuchiya, H. Gao, T. Ohmura, and K. Tsuzaki. Grain size dependence of the elastic modulus in nanostructured NiTi. *Scripta Materialia*, 63:977–980, 2010.

[67] D. Koistinen and R. Marburger. A general equation prescribing the extent of the austenitic-martensitic transformation in pure iron-carbon alloys and plain carbon steels. *Acta Metall. Mater.*, 7:59–60, 1959.

[68] H.C. Tong and C.M. Wayman. Characteristic tmperatures and other properties of thermoelastic martensites. *Acta Metall.*, 22:887–896, 1974.

[69] Z. Lekston D. Strotz, J. Palka. Nanotexture studies of NiTi shape memory alloy after severe plastic deformation with the use of TEM. *Solid State Phenomena*, 186:90–93, 2012.

[70] Y.C. Shu and K. Bhattacharya. The influence of texture on the shape memory effect in polycrystals. *Acta Mater*, 46:5457–5473, 1998.

[71] simulia. *SIMULIA Abaqus Unified FEA - Portfolio*, 2014. `http://www.3ds.com/products-services/simulia/portfolio/abaqus/overview/`.

[72] P. Snyder. tmpfs: A virtual memory file system. Sun Microsystems Inc.