Neural networks in a multiscale approach for concrete

(Neuronale Netze in einem Multiskalenansatz für Beton)

DISSERTATION

Zur Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)
an der Fakultät Bauingenieurwesen
der
Bauhaus-Universität Weimar

vorgelegt von
Dipl.-Ing. Jörg F. Unger
geb. am 25. Juni 1976 in Berlin

Weimar, März 2009

Mentor:

Prof. Dr.-Ing. habil. Carsten Könke, Bauhaus-Universität Weimar

Gutachter:

Prof. Ing. Milan Jirásek, DrSc., Czech Technical University in Prague

apl. Prof. Dr.-Ing. Wolfgang Graf, Technische Universität Dresden

Disputation: 7. Mai 2009

Abstract

From a macroscopic point of view, failure within concrete structures is characterized by the initiation and propagation of cracks. In the first part of the thesis, a methodology for macroscopic crack growth simulations for concrete structures using a cohesive discrete crack approach based on the extended finite element method is introduced. Particular attention is turned to the investigation of criteria for crack initiation and crack growth.
A drawback of the macroscopic simulation is that the real physical phenomena leading to the nonlinear behavior are only modeled phenomenologically. For concrete, the nonlinear behavior is characterized by the initiation of microcracks which coalesce into macroscopic cracks. In order to obtain a higher resolution of this failure zones, a mesoscale model for concrete is developed that models particles, mortar matrix and the interfacial transition zone (ITZ) explicitly. The essential features are a representation of particles using a prescribed grading curve, a material formulation based on a cohesive approach for the ITZ and a combined model with damage and plasticity for the mortar matrix.
Compared to numerical simulations, the response of real structures exhibits a stochastic scatter. This is e.g. due to the intrinsic heterogeneities of the structure. For mesoscale models, these intrinsic heterogeneities are simulated by using a random distribution of particles and by a simulation of spatially variable material parameters using random fields.
There are two major problems related to numerical simulations on the mesoscale. First of all, the material parameters for the constitutive description of the materials are often difficult to measure directly. In order to estimate material parameters from macroscopic experiments, a parameter identification procedure based on Bayesian neural networks is developed which is universally applicable to any parameter identification problem in numerical simulations based on experimental results. This approach offers information about the most probable set of material parameters based on experimental data and information about the accuracy of the estimate. Consequently, this approach can be used a priori to determine a set of experiments to be carried out in order to fit the parameters of a numerical model to experimental data.
The second problem is the computational effort required for mesoscale simulations of a full macroscopic structure. For this purpose, a coupling between mesoscale and macroscale model is developed. Representative mesoscale simulations are used to train a metamodel that is finally used as a constitutive model in a macroscopic simulation. Special focus is placed on the ability of appropriately simulating unloading.

Kurzfassung

Makroskopisch betrachtet kann das Versagen von Beton durch die Entstehung und das Wachstum von Rissen beschrieben werden. Im ersten Teil der Arbeit wird eine Methode zur Simulation der makroskopischen Rissentwicklung von Beton unter Verwendung von kohäsiven diskreten Rissen basierend auf der erweiterten Finiten Elemente Methode vorgestellt. Besondere Bedeutung liegt dabei auf der Untersuchung von Kriterien zur Rissentstehung und zum Risswachstum.
Ein Nachteil von makroskopischen Simulationen liegt in der nur phänomenologischen Berücksichtigung der tatsächlichen Vorgänge. Nichtlineares Verhalten von Beton ist durch die Entstehung von Mikrorissen gekennzeichnet, die bei weiterer Belastung zu makroskopischen Rissen zusammenwachsen. Um die Versagenszone realitätsnah abbilden zu können, wurde ein Mesoskalenmodell von Beton entwickelt, welches Zuschläge, Matrix und Übergangszone zwischen beiden Materialien (ITZ) direkt abbildet. Hauptmerkmal sind die Simulation der Zuschläge nach einer Sieblinie, eine kohäsive Materialformulierung der ITZ und ein kombiniertes Model aus Schädigung und Plastizität für das Matrixmaterial.
Im Gegensatz zu numerischen Simulationen ist die Systemantwort reeller Strukturen eine unscharfe Größe. Dies liegt u.a. an Heterogenitäten innerhalb der Struktur, die im Rahmen der Arbeit durch eine zufällige Verteilung der Zuschläge und über räumlich variierende Materialparameter unter Verwendung von Zufallsfeldern simuliert werden.
Zwei Hauptprobleme sind bei den Mesoskalensimulationen aufgetreten. Einerseits sind Materialparameter auf der Mesoskala oft schwer zu bestimmen. Deswegen wurde eine Methode basierend auf Bayes neuronalen Netzen entwickelt, die eine Parameteridentifikation unter Verwendung von makroskopischen Versuchen erlaubt. Diese Methode ist aber universell anwendbar auf alle Parameteridentifikationsprobleme in numerischen Simulationen basierend auf experimentellen Daten. Der Ansatz liefert sowohl Informationen über den wahrscheinlichsten Parametersatz des Models zur numerischen Simulation eines Experiments als auch eine Einschätzung der Genauigkeit dieses Schätzers. Die Methode kann auch verwendet werden, um a priori einen Satz von Experimenten auszuwählen der notwendig ist, um die Parameter eines numerischen Modells zu bestimmen.
Ein zweites Problem ist der numerische Aufwand von Mesoskalensimulationen für makroskopische Strukturen. Aus diesem Grund wurde eine Kopplungsstrategie zwischen Meso- und Makromodell entwickelt, bei dem repräsentative Simulationen auf der Mesoebene verwendet werden, um ein Metamodell zu generieren, welches dann die Materialformulierung in einer makroskopischen Simulation darstellt. Ein Fokus liegt dabei auf der korrekten Abbildung von Entlastungen.

Vorwort und Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Strukturmechanik.
Besonderer Dank gilt meinem Mentor, Prof. Carsten Könke als Initiator und Förderer meiner Projekte, der einen wesentlichen Anteil am Entstehen dieser Arbeit hat. Er hat mich immer bestmöglich unterstützt und war jederzeit ein konstruktiver Gesprächspartner, der aber auch Freiraum für eine selbständige Entwicklung gegeben hat. Seine frühzeitige Förderung der Teilnahme an internationalen Tagungen haben mich und diese Arbeit geprägt.
Vielen Dank auch an Prof. Milan Jirásek und Prof. Wolfgang Graf für die Übernahme eines Gutachtens, die wertvollen Anmerkungen und verständnisfördernden Diskussionen und die schnelle und sorgfältige Korrektur meiner Arbeit.
Herzlicher Dank gilt auch allen Kollegen am Institut für Strukturmechanik, denen ich nicht nur eine angenehme Arbeitsatmosphäre verdanke, sondern auch wesentliche Beiträge im Rahmen der gemeinsam entwickelten Software SLang.
Besonders hervorheben möchte ich in diesem Zusammenhang meinen Studien- und Bürokollegen Stefan Eckardt, der jederzeit ein wertvoller Diskussionspartner war und auch bei der praktischen Umsetzung vieler Probleme eine tatkräftige Hilfe war. Auch vielen Dank für die gewissenhafte Korrektur meiner Arbeit. Weiterer Dank gebührt Dr. Thomas Most, der mich schon als Diplomand betreut hat, und auch später einen großen Anteil u.a. auch an der thematischen Ausrichtung meiner Arbeit hatte. Auch bei der "guten Fee" Frau Terber möchte ich mich bedanken, deren Arbeit man erst dann richtig schätzen lernt, wenn sie abwesend ist. Bei Dr. Dagmar Hintze möchte ich mich für die kompetente Unterstützung bei den mir anvertrauten Lehraufgaben bedanken - auch das Hutdesign hat mir sehr gefallen. Dr. Volkmar Zabel war meine erste "Kontaktperson" am ISM, und auch der Vermittler für meine Tätigkeiten in Leuven. Weiterhin möchte ich mich bei Andrea Keßler bedanken, die in ihrer Diplomarbeit die Grundlage für die Implementation der XFEM gelegt und als Kollegin das Büroklima wesentlich bereichert hat.
Weiterhin möchte ich mich bei der Firma Dynardo, und dabei insbesondere Dr. Johannes Will und Dr. Dirk Roos, bedanken, die mir wertvolle Unterstützung bei der Einarbeitung in grundlegende stochastische Methoden gegeben haben.
Heiko Beinersdorf danke ich für die vielen kleinen Hinweise bei der Verwendung von undokumentierten Features in SLang. Auch vielen Dank an die "Tanzgruppe" für die Motivation außerhalb der Uni, obwohl das Aufstehen am nächsten Tag nicht immer einfach war. Ich danke auch meinen "Zweitprüfern", insbesondere Katrin Linne und Maik Brehm.
Ganz besonders danke ich meinen Eltern für eine wunderschöne Kindheit, die Unterstützung in allen komplizierten Situation und auch die frühzeitige Förderung, die eine grundlegende Basis dieser Arbeit darstellt. Auch vielen Dank meinen drei Brüdern Falk, Ulf und Claus, auf die ich mich immer verlassen kann.
Von ganzem Herzen möchte ich mich bei meiner Frau für die liebevolle Unterstützung bedanken, die mich auch in schwierigen Zeiten immer wieder motiviert hat. Mit unserer Tochter und dem zukünftigen Geschwisterchen sind wir ein tolles Team.

Contents

Chapter 1

Introduction

1.1 Motivation

The design of engineering structures is influenced by two conflictive principals - on the one hand, it should be safe, and, on the other hand, economical. As a consequence, a realistic and often nonlinear simulation of the structure is required for an optimization with respect to the production costs while still guaranteeing its safety.

In order to perform realistic nonlinear simulations, appropriate constitutive formulations for the applied materials have to be developed. The general approach is phenomenologically motivated. The idea is to derive a constitutive framework and then fit the free parameters to experimental data. Examples for this approach are e.g. Mises plasticity for metals or a damage formulation for concrete. These models often accurately describe the response of a macroscopic structure.

A drawback of these models is that material failure is only described phenomenologically without consideration of the real physical phenomena leading to failure. On a macroscopic level, the material is often assumed to be homogeneous. An example - extensively discussed in this thesis - is the simulation of concrete. In practical applications, concrete is modeled as a homogeneous material. Complex constitutive formulations have been derived in order to characterize its nonlinear behavior. In reality, the nonlinear behavior of concrete is characterized by the formation of microcracks which coalesce and form a macroscopic crack. The initiation of these microcrack is often triggered by the heterogenous structure of concrete on the mesoscale, where particles and matrix material can be distinguished. Furthermore, the weak zone in concrete is the interfacial transition zone as the boundary layer between matrix material and particles, which significantly influences the macroscopic response. Simulating concrete on the mesoscale allows a realistic representation of these phenomena in the numerical model. It naturally enables the explanation of certain experiments with a numerical model, e.g. a uniaxial tensile test that localizes without a prescribed weak point.

In contrast to experimental data, the observed response of a numerical model is, in general, deterministic. As a consequence, the stochastic scatter present in experimental data as well as in reality cannot be explained by the model. Engineers have tried to consider these effects by the introduction of safety factors that, on the one hand, reduce the material properties and, on the other hand, increase the loading. However, it is not always straightforward to decide which is the worst case scenario, i.e. which parameters have to be increased to obtain the highest failure probability. Furthermore, material parameters in real structures do not only vary between two similar structures of the same material, but there is also a spatial variability. Considering these effects in a numerical model enables the engineer to better approximate the reality by a numerical model.

Numerical simulations always require a model with a certain number of free parameters. In general, these parameters are determined from simple experimental tests, and afterwards, the same set of parameters is applied to the complex structural problem. An example is the determination of the compressive strength of concrete using a small cylinder or cube. In many cases, these free parameters have a real physical meaning and can directly be related to the experimental data. However, there are many problems, where the direct relation between free parameters and experimental results is not obvious. This problem is even more prominent, if several parameters interact to give the response, e.g. a test that is not uniaxial or in multiscale simulations, where often the parameters on the lower scale cannot be measured directly and only macroscopic experiments are available to determine the parameters. In this context, an automatic parameter identification procedure is required.

Another problem related to mesoscale simulations is the high computational effort. Increasing the resolution of the numerical model to lower scales, the degree of detail in the model increases and, at the same time, the complexity of the numerical model. A discretization of realistic macroscopic structures with finite elements on the mesoscale will certainly exceed the capabilities of the hardware currently available. As a consequence, methods have to be developed that couple the scales, thus allowing for the representation of lower scale effects while still keeping the computational effort manageable.

1.2 Outline and aims of the present work

The principal goal of this work is to increase the computational accuracy of numerical simulations of concrete. During the course of the work it was realized that many substantially different problems have to be solved in order to approach the global goal.

Starting from a macroscopic perspective, a discrete crack approach is applied for the simulation of crack growth in concrete structures. In this context, the extended finite element method (XFEM) is used, which allows the representation of material discontinuities and cracks independent of the underlying finite element mesh. As a consequence, remeshing in an adaptive crack growth simulation is avoided. The primary problem in discrete crack simulations is the correct determination of criteria that describe crack initiation and crack propagation, which, on the one hand, includes criteria to decide whether an existing crack grows and, on the other hand, in which direction a crack propagates. A crack growth criteria based on the minimum potential energy is developed and compared to other criteria on the basis of several examples.

In a second step, the macroscale model is refined and a mesoscale model is developed, which explicitly considers particles and mortar matrix as separate constituents according to a prescribed grading curve. Initially, the discrete crack concept of the macroscale was supposed to be applied on the mesoscale. It was realized that, due to the heterogeneities in the mesoscale, the number of discrete cracks was large and, especially, that the criteria for an adaptive calculation of evolving cracks are not suitable for heterogenous materials. As a consequence, a formulation based on a combined damage-plasticity approach is developed to simulate the nonlinear behavior of the matrix material, whereas a cohesive crack concept is used for the interfacial transition zone.

An additional aim of the work is to represent the stochastic scatter of experiments within the numerical model. This can be simulated by the intrinsic random character of the mesoscale model with a varying particle distribution. Additionally, spatially variable material parameters are introduced and modeled using correlated random fields. For this purpose, a discretization of random fields using the Fast Fourier Transform has been adapted to mesoscale models, and the influence of individual parameters such as correlation length or particle distribution on the response are investigated.
Another problem of the mesoscale models became apparent when the numerical models were to be verified with experimental results. Several mesoscale parameters have a similar influence on the macroscopic response. As a consequence, an automatic procedure for the parameter identification is required. For that purpose, Bayesian neural networks are used. An extension to multiple outputs and a full covariance matrix to describe the interaction between the errors in the approximation of multiple outputs is developed. This approach allows not only the determination of a set of parameters that fits best the experimental results, but it naturally includes predictions on the accuracy of the estimate. Furthermore, the methodology enables the design of a set of experiments to be performed in order to accurately determine the set of parameters in a numerical model.
Finally, a coupling procedure between mesoscale and macroscale model is introduced to reduce the computational effort of a full mesoscale simulation. A set of metamodels built up from neural networks and support vector machines is trained, based on a set of mesoscale simulations, to predict the response of the mesoscale model. This homogenized response is used as a constitutive formulation in a macroscale model.

Chapter 2

Discrete crack models

2.1 Introduction and state of the art

In a standard finite element formulation, discrete cracks are incorporated in the geometry formulation. The cracks are considered as external or internal boundaries and are aligned with the element edges after the meshing. This requires, for an adaptive crack growth simulation, a remeshing procedure each time a crack progresses. For large models, this procedure is computationally expensive. Furthermore, it might be impossible to fulfill certain conditions as e.g. a minimum element size or a maximum distortion of the elements in situations, where a crack passes close to the boundary or when two cracks cross each other.

The simplest way of incorporating the cohesive discrete crack approach into a finite element formulation is to use zero-thickness interface elements [1]. In general, this requires the remeshing of the finite element mesh, although for situations where the crack path is known in advance this can be avoided. An alternative is to insert interface elements between all element edges [2], but this causes mesh dependence of the results. The extended finite element method (XFEM) offers an elegant tool to model cracks within a finite element without the requirement of remeshing [3]. The inclusion of cohesive forces transmitted through the cracks is straightforward and does not require the addition of interface elements [4, 5].
An important point in discrete crack growth simulations is the criteria for determining the direction, in which an existing crack progresses. One widely used criterion is the maximum circumferential tensile strength criterion proposed by Erdogan [6]. Other common criteria are the maximum energy release rate [7] or the minimum strain energy density criteria [8]. These criteria are based on the local solution close to the crack tip. Based on the assumptions of linear elastic fracture mechanics, these criteria can be related to the local stress field at the crack tip. In order to decrease the influence of the near tip solution, which is less accurate than the far field solution, integral methods [9, 10] have been developed or nonlocal stresses at the crack tip are used [4].

In this chapter, the general features of the XFEM approach are presented. Afterwards, an automatic adaptive procedure for crack growth simulations is introduced. Several criteria for the direction of a crack extension are discussed and compared for several examples.

2.2 Representation of cracks and material interfaces using the XFEM approach

2.2.1 Introduction

The application of the extended finite element Method (XFEM), proposed by [3, 11], allows to describe cracks and material interfaces independently of the underlying mesh. This offers the possibility to perform crack growth simulations without the requirement for remeshing within each adaptation step. Furthermore, additional information concerning the functional approximation of the solution can be incorporated, e.g. close to a crack tip an ansatz based on linear elastic fracture mechanics (LEFM) describing the singularity at the crack tip can be used. In the case of a heterogeneous model with different material formulations, this approach simplifies the meshing procedure, since a regular mesh - as long as the outer boundary is regular - can be used, and the material interfaces are represented using the XFEM approach. However, the approach has the disadvantage that additional degrees of freedom are introduced. Consequently, the numerical effort for the solution procedure increases. An additional area of application are problems with a complex geometrical structure, e.g. materials where the heterogeneous microstructure is explicitly modeled, and, consequently, the meshing procedure is difficult. The XFEM approach in connection with the level set approximation offers a possibility to circumvent this mesh generation problem.

2.2.2 Displacement interpolation

In the XFEM approach, the standard displacement interpolation at a point $\boldsymbol{x}$ is enriched with an additional set of basis functions Ψ_k:

$$\mathbf{u}(\mathbf{x}) = \sum_{i=1}^{N^{tot}} \phi_i^{std}(\mathbf{x})\mathbf{u}_i + \sum_{j=1}^{N^{enr}} \sum_{k=1}^{N^f} \Psi_k(\mathbf{x})\phi_j^{enr}(\mathbf{x})\bar{\mathbf{u}}_j^k, \tag{2.1}$$

where $\phi_i^{std}(\mathbf{x})$ and $\phi_j^{enr}(\mathbf{x})$ are polynomial shape functions with a local support which satisfy the partition of unity:

$$1 = \sum_{i=1}^{N^{tot}} \phi_i^{std}(\mathbf{x}) = \sum_{j=1}^{N^{enr}} \phi_j^{enr}(\mathbf{x}). \tag{2.2}$$

It is important to mention that the shape functions ϕ_j^{enr} are not necessarily a subset of the shape functions ϕ_i^{std}, which is discussed later in section 2.2.4. Furthermore, N^{tot} is the total number of nodes, N^{enr} is the number of enriched nodes, N^f is the number of special purpose functions Ψ_k and $\mathbf{u}_i$, $\bar{\mathbf{u}}_j^k$ are nodal variables.

Within the framework of the extended finite element method, the incorporation of cohesive cracks is straightforward [4, 12]. If a cohesive crack is present, the total potential of the body

has to be modified to account for the additional cohesive forces transferred through the crack. Neglecting body forces, the total potential is given by:

$$\Pi = \int_\Omega \int_0^\varepsilon \boldsymbol{\sigma}^T(\tilde{\boldsymbol{\varepsilon}}) d\tilde{\boldsymbol{\varepsilon}}\, d\Omega - \int_{\Gamma^t} \boldsymbol{F}^T \boldsymbol{u}\, d\Gamma_t \tag{2.3}$$

$$- \int_{\Gamma^+_{coh}} \int_0^u \boldsymbol{t}_+{}^T(\tilde{\boldsymbol{u}}) d\tilde{\boldsymbol{u}}\, d\Gamma_{coh} - \int_{\Gamma^-_{coh}} \int_0^u \boldsymbol{t}_-{}^T(\tilde{\boldsymbol{u}}) d\tilde{\boldsymbol{u}}\, d\Gamma_{coh} \tag{2.4}$$

where $\boldsymbol{t}_+ = -\boldsymbol{t}_-$ are the forces per area transferred through the crack, $\boldsymbol{u}$ the displacements, Γ_{coh} the crack surface and Γ^t the boundary with prescribed tractions $\boldsymbol{F}$. Introducing the relative crack opening $\boldsymbol{w} = \boldsymbol{u}^- - \boldsymbol{u}^+$ on Γ_{coh}, the equation can be simplified to

$$\Pi = \int_\Omega \int_0^\varepsilon \boldsymbol{\sigma}^T(\tilde{\boldsymbol{\varepsilon}}) d\tilde{\boldsymbol{\varepsilon}}\, d\Omega + \int_{\Gamma_{coh}} \int_0^w \boldsymbol{t}^T(\tilde{\boldsymbol{w}}) d\tilde{\boldsymbol{w}}\, d\Gamma_{coh} - \int_{\Gamma^t} \boldsymbol{F}^T \boldsymbol{u}\, d\Gamma_t \tag{2.5}$$

using $\boldsymbol{t} = \boldsymbol{t}_+$. The crack opening $\boldsymbol{w}$ is a function of the nodal displacements $\boldsymbol{d}$ and of the difference between the corresponding shape functions for the displacement interpolation on the negative and positive side of the crack, arranged in the matrix $\bar{\boldsymbol{B}}$:

$$\boldsymbol{w} = \bar{\boldsymbol{B}}(\boldsymbol{x})\boldsymbol{d}. \tag{2.6}$$

Furthermore, the forces transferred through the crack are a function of the crack opening:

$$\boldsymbol{t} = \bar{\boldsymbol{C}}(\boldsymbol{w})\boldsymbol{w}, \tag{2.7}$$

with $\bar{\boldsymbol{C}}$ the interface material matrix. Variation of the potential in Eq. (2.5) yields the following equation system to be solved:

$$\left[\int_\Omega \boldsymbol{B}^T \boldsymbol{C} \boldsymbol{B}\, d\Omega + \int_{\Gamma_{coh}} \bar{\boldsymbol{B}}^T \bar{\boldsymbol{C}} \bar{\boldsymbol{B}}\, d\Gamma_{coh}\right] \boldsymbol{d} = \int_{\Gamma^t} \boldsymbol{N}^T \boldsymbol{F}\, d\Gamma_t, \tag{2.8}$$

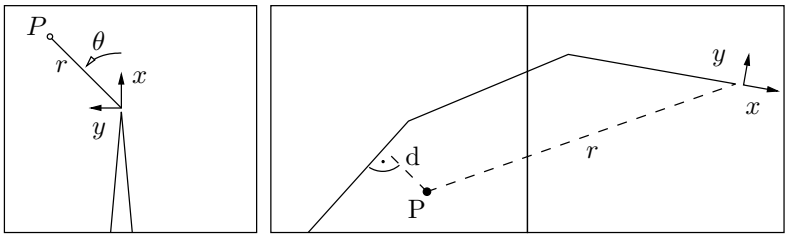
where $\boldsymbol{B}$ is the matrix relating strains and nodal displacements, $\boldsymbol{C}$ the material tensor for the bulk material and $\boldsymbol{N}$ the vector containing the shape functions for the displacement interpolation.

2.2.3 Enrichment function for cracks

Cracks are characterized by a discontinuous displacement field, and the modified Heaviside function

$$H(\mathbf{x}) = \left\{ \begin{array}{cc} -1 & \mathbf{x}\epsilon\Omega_- \\ 1 & \mathbf{x}\epsilon\Omega_+ \end{array} \right\} \tag{2.9}$$

can be used as a special purpose function Ψ_k, where Ω_-, Ω_+ are the two opposite sides of the crack faces. This approach restricts the crack to end on element edges. A more elegant way is to additionally use functions to model the crack tip. Based on the asymptotic behavior of



(a) straight crack (b) curved crack

Figure 2.1 Crack tip coordinate system and transformation for curved cracks.

the displacement field close to the crack tip in linear elastic fracture mechanics, the following special purpose functions can be used:

$$\Psi_{\alpha,\alpha=1..4} = \sqrt{r}\left[\sin\left(\frac{\theta}{2}\right), \cos\left(\frac{\theta}{2}\right), \sin(\theta)\sin\left(\frac{\theta}{2}\right), \sin(\theta)\cos\left(\frac{\theta}{2}\right)\right], \tag{2.10}$$

which accurately describe the singularity at the crack tip. However, if a cohesive model for the tractions across the crack is applied, the stress field in the vicinity of the crack tip does not possess a singularity. According to [12], the following special purpose functions for the crack tip enrichment are applied for cohesive cracks:

$$\Phi_{\alpha,\alpha=1..4} = r\left[\sin\left(\frac{\theta}{2}\right), \cos\left(\frac{\theta}{2}\right), \sin(\theta)\sin\left(\frac{\theta}{2}\right), \sin(\theta)\cos\left(\frac{\theta}{2}\right)\right], \tag{2.11}$$

where θ and r are the angle and radius of a point P in the local crack tip coordinate system, as illustrated in Fig. 2.1(a). In a general adaptive crack growth simulation with a considerable influence of mode II loading, the crack is typically curved. In order to ensure that the discontinuity in the displacement field remains on the crack trajectory, a modification such as the one proposed in [3, 13] has to be applied. In this work, a transformation is used that results in a continuous displacement field in the bulk material. The angle θ and the radius r are modified as illustrated in Fig. 2.1 with

$$\theta = \arcsin\left(\frac{d}{r}\right), \tag{2.12}$$

where d is the shortest distance to the crack and r the distance to the crack tip. For a straight crack, this approach reproduces the original functions.

For cracks which are exactly on the interface between two materials, [14] proposes the following crack tip functions:

$$\Phi_\alpha(\boldsymbol{x}) = \left\{\sqrt{r}\cos(\varepsilon\log r)e^{-\varepsilon\theta}\sin\left(\frac{\theta}{2}\right), \sqrt{r}\cos(\varepsilon\log r)e^{-\varepsilon\theta}\cos\left(\frac{\theta}{2}\right), \right. \tag{2.13}$$

$$\sqrt{r}\cos(\varepsilon\log r)e^{\varepsilon\theta}\sin\left(\frac{\theta}{2}\right), \sqrt{r}\cos(\varepsilon\log r)e^{\varepsilon\theta}\cos\left(\frac{\theta}{2}\right), \tag{2.14}$$

$$\left.\sqrt{r}\cos(\varepsilon\log r)e^{\varepsilon\theta}\sin\left(\frac{\theta}{2}\right)\sin(\theta), \sqrt{r}\cos(\varepsilon\log r)e^{\varepsilon\theta}\cos\left(\frac{\theta}{2}\right)\sin(\theta),\right. \tag{2.15}$$

$$\sqrt{r}\sin(\varepsilon\log r)e^{-\varepsilon\theta}\sin\left(\frac{\theta}{2}\right),\sqrt{r}\sin(\varepsilon\log r)e^{-\varepsilon\theta}\cos\left(\frac{\theta}{2}\right), \tag{2.16}$$

$$\sqrt{r}\sin(\varepsilon\log r)e^{\varepsilon\theta}\sin\left(\frac{\theta}{2}\right),\sqrt{r}\sin(\varepsilon\log r)e^{\varepsilon\theta}\cos\left(\frac{\theta}{2}\right), \tag{2.17}$$

$$\sqrt{r}\sin(\varepsilon\log r)e^{\varepsilon\theta}\sin\left(\frac{\theta}{2}\right)\sin(\theta),\sqrt{r}\sin(\varepsilon\log r)e^{\varepsilon\theta}\cos\left(\frac{\theta}{2}\right)\cos(\theta)\Big\}. \tag{2.18}$$

with

$$\varepsilon = \frac{1}{2\pi}\log\left(\frac{1-\beta}{1+\beta}\right) \tag{2.19}$$

$$\beta = \frac{\mu_1(\kappa_2-1)-\mu_2(\kappa_1-1)}{\mu_1(\kappa_2+1)+\mu_2(\kappa_1+1)} \tag{2.20}$$

$$\mu_i = \frac{E_i}{1+\nu_i} \tag{2.21}$$

$$\kappa_i = \frac{3-\nu_i}{1+\nu_i} \tag{2.22}$$

where μ_i, ν_i, κ_i and E_i are the shear modulus, Poisson's ratio, Kolosov constant and Young's modulus. If both materials are identical, the span of Eqs. (2.13)-(2.18) is identical to the enrichment functions for an isotropic material given in Eq. (2.10).

2.2.4 Enrichment function for material discontinuities

The displacement field for a linear bimaterial interface is continuous across the interface and has a discontinuous first derivative. Furthermore, it is preferable that the enrichment function ψ is symmetric with respect to the interface and has a constant derivative on either side of the interface so that the enrichment causes only constant strains on both sides. A reasonable choice is $\psi = |d(\mathbf{x})|$, where $d(\mathbf{x})$ is the signed distance function to the interface. Different approaches to calculate the signed distance function $|d(\mathbf{x})|$ for the special type of ellipsoidal inclusions are discussed in the following, but the results are extendible to general shapes.

2.2.4.1 Polygonal decomposition

A straightforward way to represent an interface is a decomposition into polygonal segments in 2D or polygonally bounded surfaces in 3D. This approach has the drawback that for any integration point the distance to all segments/surfaces has to be calculated (the minimum of all these distances is the signed distance function), which is - for complex geometries with a large number of segments/surfaces - computationally expensive. For an iterative procedure (e.g. to simulate discrete crack growth or a nonlinear material response) it is therefore recommendable to calculate the signed distance and its derivatives at each integration point a priori after each modification of the crack geometry and store these values, which requires additional memory. A second drawback is the fact that the approximated signed distance function does not necessarily have continuous derivatives within one element. This is due

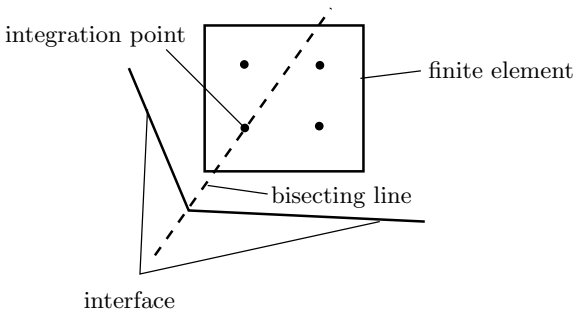


Figure 2.2 Integration point within an element on the bisecting line of two segments describing the interface.

to the fact that the distance between an integration point on the bisecting line between two segments of the polyline is identical to both segments, whereas its derivative is different, which is illustrated in Figure 2.2. This results in a discontinuous integrand in the integration of the stiffness matrix. Consequently, the integration can only be performed with a high integration order or by splitting the element into subregions, whose edges correspond to the discontinuities, which requires additional coupling between the geometry and the FE-model. Both approaches result in a high number of integration points. Furthermore, the strains and stresses are not necessarily continuous within an element even with the same material for all integration points. A third problem results from the fact that the enrichment function is not a polynomial (although the total domain can be decomposed into subsets with piecewise polynomials). This leads to problems in partially enriched elements, which are further discussed in section 2.8.1.

2.2.4.2 Approximation of the interface by level sets

A second approach is the usage of level sets for the description of the enrichment function. In [15] level sets are used to approximate the signed distance function. [5] uses two level sets to describe the geometry of a crack in 3D - one level set for the description of the crack surface and another level set orthogonal to the previous level set to describe the crack front. [16] applied the level set approach to handle complex microstructures. The advantage of the level set representation is that no background geometry model is required. Additionally, the adaptation of the mesh due to a moving interface (fluids, cracks) especially in 3D can be performed much more efficiently. [17] simulated interface material failure using the XFEM approach. In general, the idea is to define a function $f(\mathbf{x})$ within the domain Ω of the body in such a way that the subset $\Gamma = \{\mathbf{x} : f(\mathbf{x}) = 0, \mathbf{x} \epsilon \Omega\}$ coincides with the interface. This choice is not unique for a given interface (e.g. a scaling of the function $f(\mathbf{x})$ does not change the subset Γ). Since for an arbitrary interface this function is rather difficult to determine, the function $f(\mathbf{x})$ is approximated by a level set function $\bar{f}(\mathbf{x})$:

$$f(\mathbf{x}) \approx \bar{f}(\mathbf{x}) = \sum_{i=1}^{N_l} \phi_i^{lev}(\mathbf{x}) f(\mathbf{x}_i), \tag{2.23}$$

where N_l is the number of shape functions ϕ_i^{lev}, which are standard finite element shape functions. Often the same mesh is used for the interpolation of the level set function as well as for the displacements, but this is not required. For an elliptical inclusion in 2D, [15] proposed a level set function

$$f(\mathbf{x}) = \left(\frac{\xi_1(\mathbf{x})}{r_1}\right)^2 + \left(\frac{\xi_2(\mathbf{x})}{r_2}\right)^2 - 1, \tag{2.24}$$

where $\boldsymbol{\xi} = (\xi_1, \xi_2)$ are the coordinates of the point $\mathbf{x}$ in the local coordinate system of the ellipse with the origin at the center and the axes aligned with the principal axes of the ellipse. Furthermore, r_1 and r_2 are the radii in the directions of the local, principal axes. Obviously, this function is not a signed distance function. By applying the fast marching method according to [18, 19], the nodal values can be modified in an iterative procedure to obtain a signed distance function. This is equivalent to the condition that the gradient of the level set function $\nabla f(\boldsymbol{x})$ at each node i with coordinates $\boldsymbol{x}_i$ is equal to unity:

$$\bigtriangledown f(\boldsymbol{x}_i) = 1. \tag{2.25}$$

The gradient in each node is determined as the average gradient of the adjacent elements. In an iterative procedure, the stationary solution of

$$\frac{\partial \phi^{lev}(\boldsymbol{x}_i)}{\partial t} = \text{sign}\left(\phi^{lev}(\boldsymbol{x}_i)\right)\left(1 - \left|\nabla \phi^{lev}(\boldsymbol{x}_i)\right|\right) \tag{2.26}$$

is determined, where the values are initialized with the level set function. The time t is only an auxiliary variable. For numerical reasons, it is advantageous to approximate the sign-function by a smooth function as proposed by [20]:

$$\text{sign}\left(\phi^{lev}(\boldsymbol{x}_i)\right) \approx \frac{\phi^{lev}(\boldsymbol{x}_i)}{\sqrt{\phi^{lev}(\boldsymbol{x}_i)^2 + \varepsilon^2}}, \tag{2.27}$$

where ε is a small value. In the current implementation, ε was set to the minimal element size in the mesh. It is to be noted that the approximated signed distance function is almost zero for points on the isoline with $f(\boldsymbol{x}) = 0$. Consequently, only level set values of the nodes with a certain distance from the zero isoline of the level set are updated and the position of the zero isoline remains almost unchanged in the update procedure. This method has been applied to regular meshes with bilinear, quadrilateral elements. For relatively fine meshes, a good approximation of the ellipse (the interface) could be obtained. For coarse meshes (less than 4 elements in one direction in the interior of the ellipse), this method required many iteration steps, did not converge at all and/or the approximation of the interface (the ellipse) deviated considerably from the real ellipse. This is due to the fact that the gradient at each node is determined as the average of the gradient in the four adjacent elements. Since the derivatives of finite element shape functions are in general discontinuous across element boundaries and the number of elements is not sufficient to accurately approximate the level set function, strong variations of the gradients occur at the nodes.

A second possibility is to calculate the function $d(\mathbf{x}_i)$ exactly at the nodes. For an elliptical inclusion, this can only be done in an iterative way and in our case a Newton method has been applied. The details of this algorithm are given in section A.1.

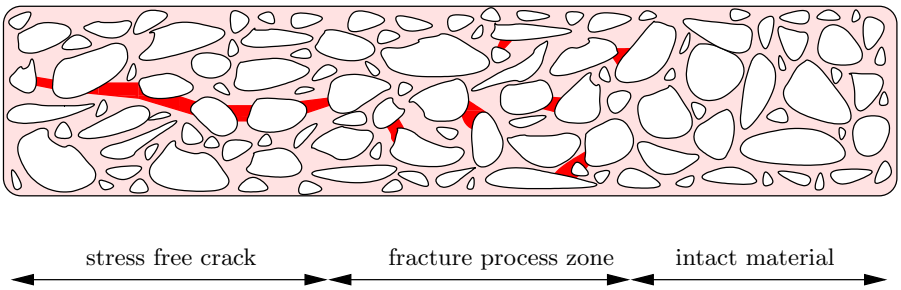


Figure 2.3 Fracture process in concrete.

The obtained representation of the interface using a direct calculation of the signed distance function is a good approximation and the method that has been finally applied. For general shapes of interfaces, it is possible to decompose the interface into polygons, if this is not the definition of the curve anyway, and calculate the distance to each segment separately. A high accuracy can be obtained, if the size of the linear subsegments is small compared to the element size used for the approximation of the level set function. It is important to mention that even by using a polygonal decomposition of the curve representing the interface, the corresponding approximated level set function $\bar{f}(\mathbf{x})$ is, within one element, as often continuously differentiable as the shape functions itself, and the derivatives can be calculated by differentiation of Eq. (2.23).

2.3 Cohesive cracks

2.3.1 Introduction

The assumption of a stress free crack with a stress singularity at the tip based on linear elastic fracture mechanics (LEFM) was first applied to concrete by Kaplan [21]. Kesler concluded that LEFM is not valid for concrete as a quasibrittle material [22], for which the fracture process is characterized by a fracture zone including an initial formation of microcracks, which coalesce during the loading process and, finally, form a macroscopic stress free crack, which is illustrated in Fig. 2.3. The size of this fracture process zone is not small compared to the specimen dimensions and, consequently, LEFM does not accurately represent the reality. In order to describe cracking phenomena in quasibrittle materials, Hillerborg developed the fictitious crack model [23]. The model is an extension of the Dugdale/Barenblatt plastic crack-tip model [24, 25], which relates normal stress and normal crack opening. It is based on the idea that, close to the crack tip, stresses between opposite faces of the crack can be transferred by mechanisms such as aggregate interlocking, friction and material bonding. The essential parameters of this model are the tensile strength of the material, which describes the maximum stress transferred through the crack at a vanishing crack opening and the fracture energy, which characterizes the amount of energy dissipated within the cohesive zone up to the point of a stress free crack. In these first approaches, only normal stresses related to the normal crack opening are considered. Different types of softening functions have been used, e.g. linear, bilinear [26] and formulations based on the exponential softening type [23].

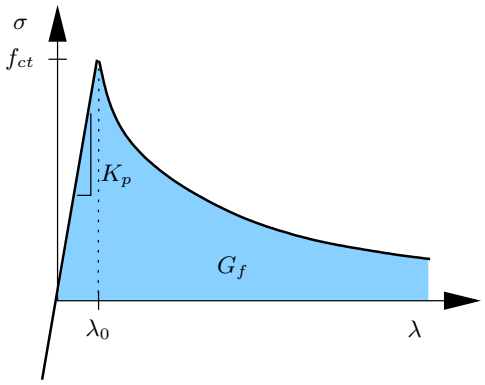


Figure 2.4 Cohesive traction-separation material law.

A number of coupled normal/shear interface laws have been subsequently developed; Vonk models the aggregate/mortar interface in concrete [27], Cervenka uses a coupled interface law for the fracture simulation of a concrete gravity damn [28] and Carol has developed a model based on a plasticity formulation [29].

2.3.2 Mixed cohesive zone model

In the current work, a model based on [30, 31] is used, which describes normal as well as tangential tractions along an interface. A total crack opening λ is introduced:

$$\lambda = \sqrt{\langle u_n^2 \rangle + (\alpha u_t)^2}, \tag{2.28}$$

where u_n and u_t are the normal opening and tangential sliding of the interface surfaces, and α is a material constant which controls the weighting between the normal and tangential opening. Note that by using the McCauley brackets, only a positive normal crack opening contributes to the total crack opening. Furthermore, a cohesive traction-separation law is used for loading conditions:

$$\sigma(\lambda) = \left\{ \begin{array}{cc} K_p \lambda & \lambda < \lambda_0 \\ f_{ct} e^{\frac{-f_{ct}(\lambda - \lambda_0)}{G_f}} & \text{otherwise} \end{array} \right\} \tag{2.29}$$

where $\lambda_0 = f_{ct}/K_p$ is the crack opening at which the linear elastic peak load is reached, K_p is the penalty stiffness, f_{ct} the tensile strength of the interface layer and G_f its fracture energy as illustrated in Fig. 2.4. The penalty part is important to allow a crack to remain closed for compressive stresses and small tensile stresses.

Assuming that a potential Φ

$$\Phi(u_n, u_t) = \int_0^{\lambda(u_n, u_t)} \sigma(\lambda') d\lambda' \tag{2.30}$$

exists, the normal and tangential tractions are obtained respectively by:

$$T_n = \frac{\partial \Phi(u_n, u_t)}{\partial u_n} = \sigma(\lambda)\frac{u_n}{\lambda} \tag{2.31}$$

$$T_t = \frac{\partial \Phi(u_n, u_t)}{\partial u_t} = \sigma(\lambda)\frac{\alpha^2 u_t}{\lambda}. \tag{2.32}$$

Describing the total potential as a function of the mixed displacement λ leads to the assumption that the fracture energies for mode I and mode II are equivalent, although this differs from the general assumption that the mode II fracture energy for concrete is higher than the mode I fracture energy. However, an advantage for the numerical implementation is that the obtained stiffness matrix is symmetric. The penalty stiffness K_p must be chosen carefully. On the one hand, the penetration of the two adjacent crack faces of the interface in compression has to be reduced by making K_p as large as possible. On the other hand, a high penalty stiffness results in an ill-conditioned global stiffness matrix. In the implementation of the model, the penalty stiffness was determined using an empirical approach.

A history variable λ_{max}, which corresponds to the maximum total crack opening λ reached during loading, is required to decide, whether the material point is under loading ($\lambda \geq \lambda_{max}$) or unloading($\lambda < \lambda_{max}$) conditions. A damage model is applied for the unloading path, and a linear function back to the origin is assumed:

$$T_n = \sigma(\lambda_{max})\frac{u_n}{\lambda_{max}} \tag{2.33}$$

$$T_t = \sigma(\lambda_{max})\frac{\alpha^2 u_t}{\lambda_{max}}. \tag{2.34}$$

If the interface is in compression, the contact condition is approximated by the penalty stiffness

$$T_n = K_p u_n, \tag{2.35}$$

and the parameter λ is only a function of the tangential displacement

$$\lambda = |\alpha u_t|. \tag{2.36}$$

2.4 Crack initiation criteria

In this work, the initiation of cracks is modeled by the principal stress criterion. If the principal stress after each equilibrium step exceeds the tensile strength of the material at any integration point, an additional crack is introduced, orthogonally to the corresponding eigenvector. The introduced additional crack is centered at the integration point and has a prescribed length, which is chosen in the order of two to three times the element length to avoid the existence of two crack tips of the same crack within a single element. Additionally, the intersection with the external boundary of the model is verified and, if necessary, the crack tip is moved to the intersection. In the latter case, the enrichment with the Heaviside function is used instead of the crack tip functions.

2.5 Crack growth criteria

The decision, whether an existing cohesive crack grows, can be determined by various methods published in literature. In [32], a Virtual Crack Extension(VCE) method is introduced, which extends the total potential from Linear Elastic Fracture Mechanics to cohesive forces. In [33] an extension for plastic material properties is proposed. However, this virtual crack extension approach has not been followed due to numerical problems. In general, the energy release rate with respect to a small crack extension Δa is calculated using finite differences. In the XFEM approach, a small extension of the crack is difficult, since either a short segment of length Δa has to be introduced at the crack tip, which requires remeshing of the integration cells with small element edges, or the segment with the crack tip has to be prolongated, which requires a mapping of the history variables for the integration points on that segment and modifies the full displacement field, since the enrichment function depends on the geometric representation of the crack.

In this work, a criterion based on the stress state of the cohesive crack tip is used. A crack is assumed to progress, if the stress within the cohesive interface at a certain distance from the crack tip exceeds the elastic limit. In the implementation, the reference point is chosen as the closest integration point within the cohesive crack with a distance of a quarter of the last crack increment to the crack tip. Note that, due to the intersection with the element edges, the actual location of the reference point slightly depends on the underlying mesh. If this reference point is within the inelastic domain corresponding to $\lambda > f_{ct}/K_p$ for the applied material model, the crack is extended with a prescribed extension length. This choice is motivated by the fact that the reference point should be close to the crack tip in order to ensure that the last crack segment is almost entirely in the elastic domain. In order to avoid locking effects when the intersection of the last crack segment and the element containing the crack tip is very short, a certain minimum distance to the crack tip is required. The choice of a quarter of the last crack increment to the crack tip is obtained empirically. Due to the penalty approach for the interface law, it is still possible that a newly inserted crack segment remains closed. As a result, it is not critical that a crack might be extended slightly too early.

The prescribed length of the new segments depends on the curvature of the expected crack trajectory. A curved crack is approximated by a polygon, with the length of each segment of the polygon being the prescribed extension length. For larger curvatures of the cracks more and shorter segments are required to accurately approximate the crack.

2.6 Direction of crack extension

The direction of the crack extension plays an important role in the numerical simulation. This is due to the fact that oscillations around the correct crack path caused by an inaccurate determination of the direction of the crack extension lead to a larger crack surface, which may result in locking phenomena. Furthermore, incorrect extension directions lead to significant changes in the global load-displacement curve. Various criteria for the direction of a crack extension have been proposed in literature; these include maximum circumferential tensile stress [6], maximum energy release rate [7] and minimum strain energy density criteria [8].

In the context of the XFEM, Wells has proposed a criterion based on the nonlocal maximum principal stress [4]. A discussion on the drawbacks related to this criterion is given in section 2.9.2. Another popular criterion is the global tracking algorithm [34]; here the local vector field of the extension criterion at certain points (e.g. maximum principal stress and corresponding eigenvector) is used to obtain a global vector field, which can be used for the interpolation of the direction criterion at the crack tip.

2.6.1 Maximum circumferential stress

Erdogan has proposed a crack extension criterion, in which the crack extends in the direction of maximum circumferential tensile stress [6]. Assuming a stress-free crack, the near tip displacements around the tip in a polar crack tip coordinate system can be described by the stress intensity factors K_I and K_{II} by

$$u_I = K_I \frac{1}{2\mu}\sqrt{\frac{r}{2\pi}}\cos\left(\frac{\theta}{2}\right)\left[\kappa - 1 + 2\sin^2\left(\frac{\theta}{2}\right)\right] + K_{II}\frac{1}{2\mu}\sqrt{\frac{r}{2\pi}}\sin\left(\frac{\theta}{2}\right)\left[\kappa + 1 + 2\cos^2\left(\frac{\theta}{2}\right)\right] \tag{2.37}$$

$$u_{II} = K_I \frac{1}{2\mu}\sqrt{\frac{r}{2\pi}}\sin\left(\frac{\theta}{2}\right)\left[\kappa + 1 - 2\cos^2\left(\frac{\theta}{2}\right)\right] - K_{II}\frac{1}{2\mu}\sqrt{\frac{r}{2\pi}}\cos\left(\frac{\theta}{2}\right)\left[\kappa - 1 - 2\sin^2\left(\frac{\theta}{2}\right)\right], \tag{2.38}$$

where $\kappa = \frac{3-\nu}{1+\nu}$ for plane stress, $\kappa = 3-4\nu$ for plane strain and $\mu = \frac{E}{2(1+\nu)}$. By substitution of the crack opening displacements near the tip from the numerical model into Eqs. (2.37) and (2.38), virtual - there being no singularity for a cohesive crack - stress intensity factors can be calculated. The maximum circumferential tensile stress criterion finally gives the direction of the crack extension:

$$\tan\left(\frac{\theta}{2}\right) = \frac{1}{4}\frac{K_I}{K_{II}} \pm \frac{1}{4}\sqrt{\left(\frac{K_I}{K_{II}})^2 + 8\right)}. \tag{2.39}$$

By using this criterion, it is implicitly assumed that the cohesive forces do not alter the direction of the crack extension, similar to [5, 35].
This method requires an additional elastic computation, which does not consider the effect of cohesive forces. Compared to the number of iterations performed to obtain equilibrium (with cohesive effects), these additional solution steps are not time-consuming. In order to reduce the influence of local errors in the solution, the stress intensity factors in Eq. (2.37) are calculated as an average from different points close to the crack tip. Using this approach, four points on the last crack segment (at 1/8, 3/8, 5/8, 7/8 from the crack tip in local crack segment coordinates, with 0 being the tip and 1 the opposite end) have been examined and the average of these values is used. Although the displacement correlation technique does not yield accurate stress intensity factors, it is only the ratio between K_I and K_{II} that is important for the calculation of the crack extension, which seemed to be more accurate in the numerical examples.

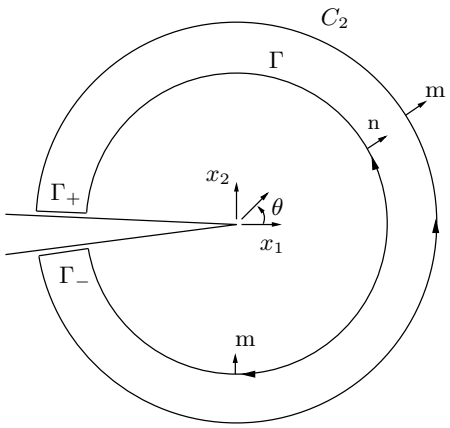


Figure 2.5 Calculation of path-independent J-integral.

2.6.2 Maximum energy release rate

Another way of describing the stress state at the crack tip is the utilization of domain integrals such as J_1 and J_2. The J-integrals are defined as:

$$J_k = \int_\Gamma \left[w\, n_k - \sigma_{ij} \frac{\partial u_i}{\partial x_k} n_j \right] d\Gamma, \tag{2.40}$$

with Γ an arbitrary closed path around the crack tip singularity and w the strain energy density. The evaluation of the boundary integral in Eq. (2.40) is somewhat cumbersome since, for a numerical implementation, the strains/stresses are not continuous along element edges. For this reason, a transformation into an area integral is recommended [9].
By using the following weighting function q:

$$q = \left\{ \begin{array}{ccc} 1 & \text{on} & \Gamma \\ 0 & \text{on} & C_2 \\ \text{arbitrarily} & \text{elsewhere} & \end{array} \right\}, \tag{2.41}$$

Equation (2.40) can be written as the integral around the closed contour Γ^*, with $\Gamma^* = \Gamma + C_2 + \Gamma_+ + \Gamma_-$. Note that $m = -n$ on Γ, which is illustrated in Fig. 2.5. For stress free cracks, this gives:

$$J_k = \int_{\Gamma^*} \left[\sigma_{ij} \frac{\partial u_i}{\partial x_k} q\, m_j - w\, q\, m_k \right] d\Gamma^*. \tag{2.42}$$

By applying the divergence theorem, the contour integral can be rewritten as the surface integral [9]:

$$J_k = - \int_{A^*} \left[w \delta_{jk} - \sigma_{ij} \frac{\partial u_i}{\partial x_k} \right] \frac{\partial q}{\partial x_j} dA^*, \tag{2.43}$$

where A^* is the surface enclosed by Γ^*. In the implementation of the method, all elements within a user-defined radius of the crack tip are selected. Furthermore, the weighting function

q for all the nodes on the boundary of that region are set at zero, whilst a value of $q = 1$ is assigned to all the inner nodes. The function q is interpolated by using bilinear shape functions within the quadrilateral elements. As a result, only the elements on the outer boundary of the selected elements contribute to the integral, since in all other elements $\frac{\partial q}{\partial x_j} = 0$. The user-defined radius is chosen as 2 to 3 times the length of the diagonal within the crack tip element. As with the displacement correlation approach, the domain integrals are evaluated in a second calculation, in which cohesive forces are ignored.
Using the formula by [36], the energy release rate can be expressed as a function of J_1 and J_2:

$$G = J_1 \cos(\theta) + J_2 \sin(\theta), \tag{2.44}$$

Maximization of Eq. (2.44) yields

$$\theta = \arctan\left(\frac{J_2}{J_1}\right), \tag{2.45}$$

from which the angle of the crack extension is determined.

2.6.3 Minimization of total potential

Both the displacement correlation technique and the J-integral method have the disadvantage that the cohesive crack progresses in the same direction as a stress-free crack. In order to investigate the influence of the cohesive forces on the direction of a crack extension, an algorithm based on the maximum energy released from the system has been developed, which is based on an idea from [37] to model crack growth in concrete with the XFEM approach. In general, the first law of thermodynamics assuming an adiabatic system (no heat exchange) and quasi-static conditions (no kinetic energy), can be written as:

$$\frac{\partial W}{\partial t} = \frac{\partial (U^e + U^p)}{\partial t} + \frac{\partial (\Gamma^e + \Gamma^d)}{\partial t}. \tag{2.46}$$

This equation represents the energy balance during crack growth, where W is the work of the external forces, U^e/U^p the elastic/plastic strain energy, Γ^e the elastic energy stored in the crack and Γ^d the energy dissipated within the cohesive crack. It implies that the energy rate introduced into the system by the external loads is equal to the sum of the internal strain energy rate and the energy rate dissipated during crack growth. It is assumed that the crack progresses in the direction, in which the potential energy of the system $(W - U^e - \Gamma^e)$ has a minimum, or likewise that the dissipated energy $(U^p + \Gamma^d)$ has a maximum.

Starting from the initial state in equilibrium, the crack is extended in a certain direction, whilst the applied loads and displacements are kept constant. The equilibrium state of the system is determined (using a Newton-Raphson iteration) and the work of the external forces and the internal elastic energy is calculated. For Dirichlet boundary conditions, the work of the external forces vanishes, whereas for Neumann boundary conditions, the external work ΔW_{ext} is obtained by

$$\Delta W_{ext} = \sum_{i=1}^{n} F_i(\Delta u_i), \tag{2.47}$$

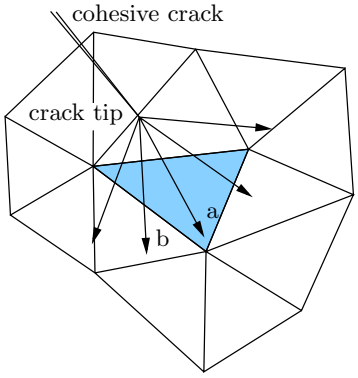


Figure 2.6 Extension of a crack in various directions as part of the search for the maximum energy release rate

where F_i is the applied nodal force and Δu_i the variation of the displacements due to an extension of the crack in a certain direction. The internal elastic energy can be determined in a similar way:

$$U^e = \frac{1}{2} \int_\Omega \boldsymbol{\sigma}(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon_p}) : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon_p}) \; d\Omega \tag{2.48}$$

$$\Gamma^e = \frac{1}{2} \int_\Gamma \boldsymbol{\sigma}(\boldsymbol{w}) : \boldsymbol{w} \; d\Gamma, \tag{2.49}$$

where $\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon_p}$ is the elastic strain, $\boldsymbol{\sigma}$ the corresponding stress in the bulk material and $\boldsymbol{w}$ the relative crack opening with the cohesive stress $\boldsymbol{\sigma}(\boldsymbol{w})$ transferred through the crack surface Γ. In the presented examples, the bulk material is linear elastic, which implies that plastic strains and, consequently, the plastic strain energy vanish.

Using this procedure, the dissipated energy can be determined for a discrete number of directions. In order to determine the direction with the maximum dissipated energy, an approximated function using these discrete directions as support points is calculated, from which the maximum is determined. A straightforward option is a polynomial regression. One disadvantage is that the approximation does not have a local character, which means that not only support points close to the point itself determine its approximated value. A better option is a MLS (moving least squares) approximation. Detailed information about the applied implementation are given in [38]. For the presented examples, a linear base polynomial with Gaussian weighting functions has been used.

A problem arises for crack extensions, which are still partially closed. Particularly, when using irregular triangular meshes, the energy dissipated within cracks depends on the length of the crack extension which actually opens. A crack within an element is in general either in the elastic or in the inelastic domain. The problem is illustrated in Fig. 2.6. Crack extension a) passes through two elements. This means that if the applied loading is sufficiently great, the integration points on the crack within the first element (with the current crack tip) exceed the tensile strength of the interface, and the crack opens within this element. If the loading is even higher, the crack opens/exceeds the elastic limit in all segments up to the new crack tip. Otherwise, some of the segments remain closed (within the elastic region). That means that for a partially closed crack extension the energy that can be dissipated strongly depends on

the position and the number of elements cut by a crack extension. Crack extension b) has a much shorter segment in the shaded element, and as a result a lower global load is required to exceed the tensile strength of the interface. Finally this implies that for this global load, the energy dissipated for extension b) is higher compared to a). This mesh bias can be avoided if the load is increased up to a point, at which, for almost all possible directions of crack extensions, the cracks are fully open. From a computational point of view, it is too time-consuming to increase the load until all extensions are fully open, and, furthermore, it is not clear whether all extensions open at all. In a simplified approach, only the straight extension (no kink of the crack) is verified in order to determine the load level of the external loads to be used for the calculation of the direction. The load is increased until the elastic limit is exceeded for a certain reference point (in this case 3/4 of the straight crack extension, similar to the crack extension criteria). For this load level, the minimum of the total potential for all directions with a constant prescribed length of the extension is calculated, which corresponds to the length of the crack extension.
Since the (nonlinear) system of equations has to be solved for each possible direction of a crack extension, the computational effort is much higher compared to the previous methods, but it can be effectively parallelized. The total number of additional (nonlinear) solution steps is given as the product of the number of crack extensions (which correlates with the prescribed length of a crack extension) and the number of different directions verified for each single crack direction to determine the minimum of the total potential. Additionally, for each crack extension, the load level has to be determined, which usually requires 2 to 3 solution steps for each single crack extension. The computational effort can further be reduced if, by using one of the previously presented methods, an estimate of the direction of crack extension is calculated, and the direction search is only performed close to that estimate; consequently the number of search directions can be substantially reduced.

2.7 Mesh refinement using the Quadtree data structure

As described in the previous section, the X-XFEM offers an efficient tool for modeling cracks without the requirement for the element edges to coincide with the discontinuity. Starting with an initial coarse mesh, an efficient refinement strategy is required which increases the node density close to the crack tip as illustrated in Fig. 2.7. This can be achieved by using the Quadtree data structure (and for 3D applications the Octree data structure). A comprehensive overview can be found in [39]. Starting with an initial mesh, the elements are iteratively decomposed into 4 subelements until the required accuracy is achieved. The shape of the initial elements determines the shape of their subelements; thus a square is decomposed into 4 squares and a rectangle into 4 rectangles. Using this approach, any polygonally bounded domain can be refined, if an initial mesh can be created. The required accuracy of the refinement is correlated with the prescribed extension length of the crack. Although a curved crack can be represented within a single enriched element, the angle of the crack kink within an element should be restricted. This is due to the fact that, within an enriched element, there is only a constant number of additional degrees of freedom to describe the displacement field, and the more complex the crack geometry is, the less accurate the solution will be. A good choice is to use the same size for the prescribed extension length and the maximum

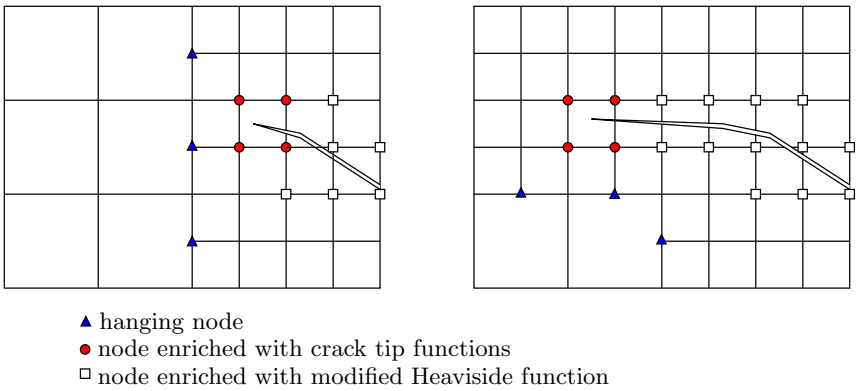


Figure 2.7 Mesh adaptation for a crack extension.

element dimension of the elements on the lowest leaf of the quadtree refinement. All elements that are cut by the crack extension are iteratively decomposed into 4 subelements, so that all elements cut by the crack finally have the same refinement level. The general procedure for the automatic crack growth algorithm is summarized in Algorithm 2.1.

Furthermore, all the nodes in the element containing the crack tip are enriched with crack tip functions, whilst all other nodes in fully cracked elements are enriched with the modified Heaviside function. This is illustrated in Fig. 2.7.

In the general case of nonlinear material behavior, the mesh adaptation requires a mapping procedure for the history variables for the transfer from the original to the adapted mesh. In [40], various approaches for the mapping are compared. It has been found that a closest-point transfer (data from the closest point of the old mesh is assigned to the new point) and a least-squares interpolation using the integration points of the old mesh are suitable. On the other hand, a shape-function projection, mapping the data from the old mesh to the nodes of the new mesh and from there to the new integration points, leads to an artificial damage diffusion.

If two neighboring elements are not on the same refinement level, additional effort is required. Sukumar proposes a mapping of regular elements with 5 to 7 nodes positioned equally spaced on the unit circle [41]. Within these elements, the natural neighbor interpolation is used, which is linear between nodes on the boundary, and, as a result, the coupling with linear finite elements is compatible. This procedure gives accurate results, even when using a Gauss quadrature for the integration of the stiffness matrix, although the interpolant is not a polynomial. However as explained in the next section, higher order elements are required; this would lead to an incompatible displacement interpolation (quadratic elements with linear natural neighbor interpolation). A second option, applied in this work, is to set up additional constrained equations, which couple the "hanging nodes" with the nodes of the adjacent coarser element. It is further recommended to limit the difference in the refinement level between two neighboring elements to one in order to obtain a smooth variation of the node density within the domain. By applying this quadtree refinement procedure, the number of degrees of freedom of the structure can be considerably reduced without loss of accuracy.

Algorithm 2.1 Automatic crack growth algorithm

```
Create initial mesh with quadrilateral elements.
Initialize quadtree structure with existing elements as root elements.
for (load=0 to final loadstep) do
  Compute equilibrium state of current loadstep.
  while (any existing crack meets the criteria for crack extension) do
    Determine the direction of crack extension.
    Adapt the mesh using the quadtree decomposition.
    Extend the crack.
    Compute equilibrium state of current loadstep.
  while any integration point meets the criteria for crack initiation do
    Determine the crack tips of the new crack.
    Adapt the mesh using the quadtree decomposition.
    Add the crack.
    Compute equilibrium state of current loadstep.
  Update history variables.
```

2.8 Numerical implementation

2.8.1 Choice of the shape functions for standard and enriched interpolation

In general, the shape functions $\phi_i^{std}(\mathbf{x})$ and $\phi_j^{enr}(\mathbf{x})$ can have a different polynomial order, but often linear functions are used. Stazi proposed to use higher order polynomials [42] in order to increase the convergence rate. Chessa demonstrated that if the enrichment function Ψ is a piecewise continuous polynomial of order n (as e.g. the signed distance function for a straight interface), the orders p^{std} and p^{enr} of the shape functions ϕ_i^{std} and ϕ_j^{enr} have to fulfill the condition $p^{std} \geq p^{enr} + n$ [43]. This is due to the fact that only fully enriched elements fulfill the partition of unity for the enriched shape functions $\phi_j^{enr}(\mathbf{x})$ and the enrichment function Ψ can exactly be represented. However, in partially enriched elements, where only certain nodes are enriched, the summand $\Psi(\mathbf{x})\phi_j^{enr}$ is at least quadratic (since the shape functions are at least linear and the signed distance function for an interface in partially enriched elements is also almost linear). Due to the fact that the enriched shape functions do not fulfill the partition of unity, a linear interpolation corresponding to a constant strain field can only be obtained if the standard shape functions are at least quadratic.
Using level sets for the interpolation of the signed distance function as the enrichment function Ψ, the order of the required interpolation p_{std} can be calculated exactly, if p_{enr} is given and the same mesh for the interpolation of level sets and displacements is used.
In this work, quadrilateral elements with 9 nodes for the standard interpolation (ϕ_{std}) are used. For the interpolation of the enriched degrees of freedom and the level set functions only the 4 corner nodes with bilinear interpolation are used, which is illustrated in Fig. 2.8.

$$\text{span}(\phi^{std}) = \left\{1, x, y, x^2, xy, y^2, x^2y, xy^2, x^2y^2\right\} \tag{2.50}$$

$$\text{span}(\phi^{enr}) = \text{span}(\phi^{lev}) = \{1, x, y, xy\} \tag{2.51}$$

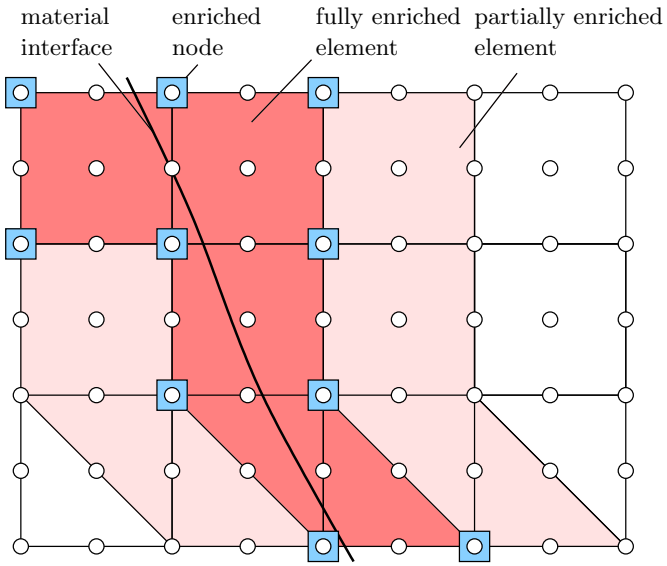


Figure 2.8 Enrichment of nodes with additional DOF's for a material interface.

Obviously, $\text{span}(\phi^{enr}) \oplus \text{span}(\phi^{lev}) = \text{span}(\phi_{std})$, and, as a result, linear displacement fields can be represented also within partially enriched elements.

An analog approach is used for triangular elements with a quadratic interpolation for the standard interpolation, where all 6 nodes are used, whereas the shape functions for the enriched DOF's are restricted to the linear shape functions of the triangle using only the corner nodes.

2.8.2 Restraining additional DOF's

A numerical problem arises, if an additional shape function is linear dependent on the standard shape functions. This happens e.g., if the enrichment function is a linear combination of standard shape functions. The problem is illustrated in Fig. 2.9. According to [44], an enrichment function has to be removed from a node, if either the area of support above the

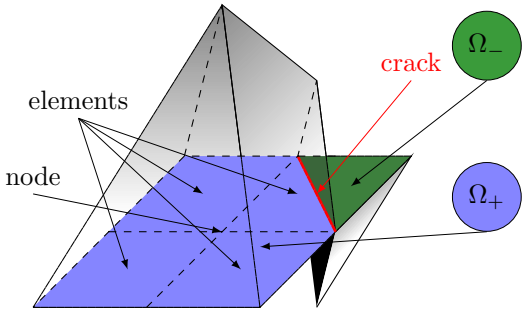


Figure 2.9 Additional shape function of a node enriched with the Heaviside function.

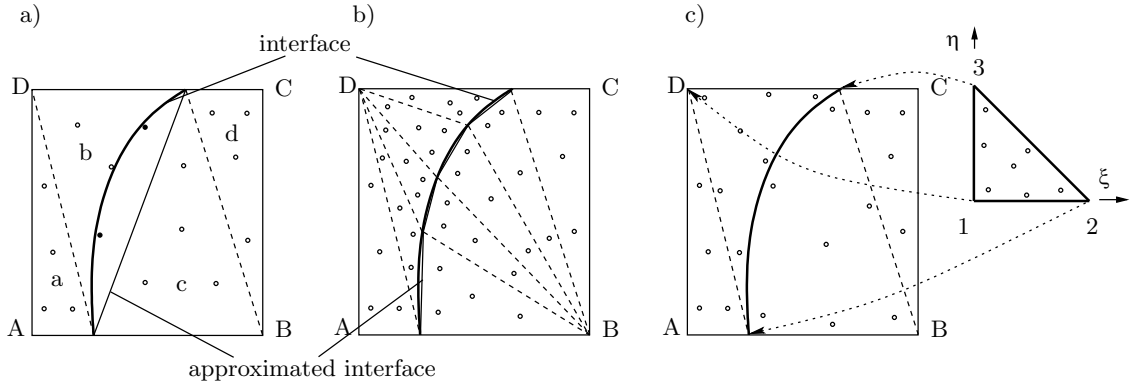


Figure 2.10 a)minimal polygonal approximation of the level set b) approximation using 4 segments and c) mapping of the unit triangle to a triangle with a curved edge.

crack $A(\Omega_+)$ or the area of support below the crack $A(\Omega_-)$ is small compared to the total area of support:

$$\frac{A(\Omega_+)}{A(\Omega_+)+A(\Omega_-)} < \text{tolerance} \qquad \text{or} \quad \frac{A(\Omega_-)}{A(\Omega_+)+A(\Omega_-)} < \text{tolerance}. \tag{2.52}$$

In order to calculate the area of support, the integration cells of each element used for the numerical integration of the stiffness matrix have been used. In this work, a tolerance of 10^{-4} has been used.

2.8.3 Integration of the system matrices

The integration of the stiffness matrix in Eq. (2.8) is performed numerically. If nonpolynomial enrichment functions Ψ as e.g. crack tip functions are used, a Gauss quadrature is only an approximation. As proposed by many authors, e.g. [11], a decomposition into triangular integration zones is performed. This can be done exactly, if the crack/interface is a polygon. When using level sets in combination with higher order elements (higher than linear triangular elements), the interface is represented by the zero isoline of a nonlinear level set function. Sometimes, many segments have to be introduced to approximate this nonlinear function by polygonal segments. Furthermore, it has to be assured that all integration points are on the "right" side, otherwise considerable errors in the numerical integration are obtained. This situation is illustrated in Fig. 2.10 with a 4-point Gauss quadrature per triangle. The approximation of the crack/interface in triangle a is not sufficient and integration points are on both sides of the interface.

In order to circumvent this problem, a new transformation is proposed that maps a triangular integration zone in the element coordinate system to a triangle with a curved interface in the global coordinate system that corresponds to the zero isoline of the level set. A transformation for a bilinear quadrilateral element is described in the following. The level set function f is

interpolated according to Eq. (2.23). For a bilinear interpolation, the zero isoline of the level set function can be described by

$$0 = a + bx + cy + dxy, \tag{2.53}$$

where the constants a,b,c,d depend of the level set values v_i of the nodes.

$$a = (v_A + v_B + v_C + v_D)/4 \tag{2.54}$$
$$b = (v_A - v_B - v_C + v_D)/4 \tag{2.55}$$
$$c = (v_A + v_B - v_C - v_D)/4 \tag{2.56}$$
$$d = (v_A - v_B + v_C - v_D)/4 \tag{2.57}$$

Without loss of generality, it is assumed that point 1 of the triangle with the coordinates in the unit system (0,0) is mapped to the point that is not located on the zero isoline. The transformation should be linear between points 1-2 and 1-3 and the following ansatz is made for the transformation:

$$x(\xi,\eta) = x_1 + (x_2 - x_1)\xi + (x_3 - x_1)\eta + \xi\eta g_x(\xi) \tag{2.58}$$
$$y(\xi,\eta) = y_1 + (y_2 - y_1)\xi + (y_3 - y_1)\eta + \xi\eta g_y(\xi). \tag{2.59}$$

$g_x(\xi,\eta)$ and $g_y(\xi,\eta)$ are functions that are to be determined later, x_i, y_i are the coordinates of the vertices of the triangle in the element coordinate system and ξ,η are the coordinates in the unit triangle. The dependence g_i on ξ is chosen randomly and, in an analog way, $g_i(\eta)$ would be possible. Obviously, the transformation is linear between points 1-2 and 1-3, since the term $g_i(\xi)$ is canceled out by the factor $\xi\eta$. The edge $\xi + \eta = 1$ between the points 2-3 should be mapped so that all points fulfill the condition in Eq. (2.53). Substitution of Eqs. (2.58) and (2.59) in Eq. (2.53) gives

$$0 = P_1 + P_2 g_x(\xi, 1-\xi) + P_3 g_y(\xi, 1-\xi) \tag{2.60}$$
$$+ P_4 g_x(\xi, 1-\xi) g_y(\xi, 1-\xi), \tag{2.61}$$

where the variables P_i are functions of ξ, x_i, y_i.

$$\begin{aligned} P_1(\xi) =& a + b\left[x_1 + (x_3 - x_1)(1-\xi) + (x_2 - x_1)\xi\right] \\ &+ c\left[y_1 + (y_3 - y_1)(1-\xi) + (y_2 - y_1)\xi\right] \\ &+ d\left[x_1 + (x_3 - x_1)(1-\xi) + (x_2 - x_1)\xi\right] \\ &\cdot \left[y_1 + (y_3 - y_1)(1-\xi) + (y_2 - y_1)\xi\right] \end{aligned} \tag{2.62}$$
$$P_2(\xi) = \xi(1-\xi)\left[\{y_1 + (1-\xi)(y_3 - y_1) + \xi(y_2 - y_1)\}\, d + b\right] \tag{2.63}$$
$$P_3(\xi) = \xi(1-\xi)\left[\{x_1 + (1-\xi)(x_3 - x_1) + \xi(x_2 - x_1)\})d + c\right] \tag{2.64}$$
$$P_4(\xi) = d\xi^2(1-\xi)^2. \tag{2.65}$$

If $P_2(\xi) \neq 0\ \forall \xi\, \epsilon\, [0,1]$, a possible choice for $g_i(\xi)$ is

$$g_x(\xi) = -\frac{P_1(\xi)}{P_2(\xi)} \tag{2.66}$$
$$g_y(\xi) = 0 \tag{2.67}$$

and analog for $P_3(\xi) \neq 0 \; \forall \xi \, \epsilon \, [0,1]$

$$g_x(\xi,\eta) = 0 \tag{2.68}$$

$$g_y(\xi,\eta) = -\frac{P_1(\xi)}{P_3(\xi)}. \tag{2.69}$$

Substituting Eqs. (2.54)-(2.57) into Eq. (2.63), and using the condition that $x_i, y_i \in [-1,1]$, the condition for $P_2 = 0$, which means that the transformation in Eq. (2.66) is not valid, is

$$d = v_A - v_B + v_C - v_D \begin{cases} > 0 \;\; \Rightarrow \;\; v_A > v_B, \; v_C > v_D \\ < 0 \;\; \Rightarrow \;\; v_A < v_B, \; v_C < v_D \end{cases} . \tag{2.70}$$

Otherwise the transformation in Eq. (2.66) can be used. Similarly, a condition for the second transformation in Eq. (2.68) can be derived:

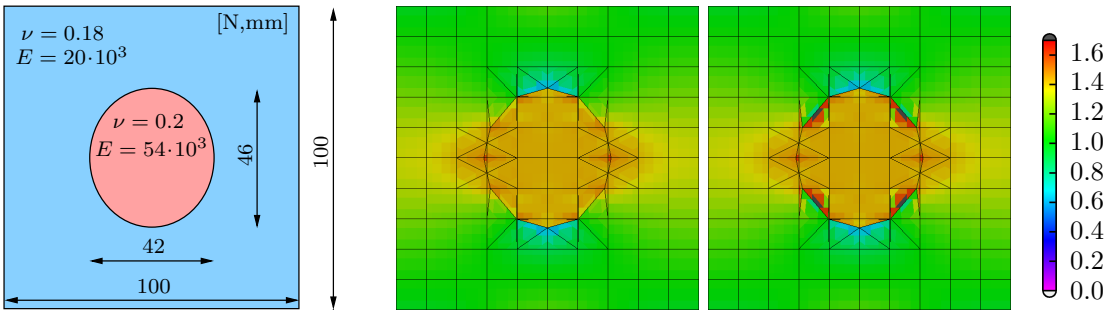
$$d = v_A - v_B + v_C - v_D \begin{cases} > 0 \;\; \Rightarrow \;\; v_A > v_D, \; v_C > v_B \\ < 0 \;\; \Rightarrow \;\; v_A < v_D, \; v_C < v_B \end{cases} . \tag{2.71}$$

The case, where both conditions in Eqs. (2.70) and (2.71) are fulfilled which means that none of the two transformations are valid, is, from a practical point of view, irrelevant, since the underlaying level set function then does not correspond to a signed distance function of a polygon. That would e.g. for $d > 0$ mean that $v_A > v_B, v_A > v_D, v_C > v_D, v_C > v_B$, which corresponds to a roof like function.

The triangulation within the natural coordinate system is based on a Delaunay triangulation, where the software package 'triangle' [45] has been used.

The influence of the mapping is demonstrated for a specimen with a hard inclusion in a soft matrix in Fig. 2.11. For the standard mapping without the transformation, spurious stress concentrations are obtained in the elements, which are cut by the crack, especially in the inclusion with the higher Young's modulus. However, the overall stress distribution coincides well for both methods. The mapping procedure has the disadvantage that only an element can be cut by single crack. In the general case with the distance between inclusions smaller than the element size, it is recommended to combine the two methods - a mapping is performed for the majority of the cracked elements which have only a single crack, whereas the standard integration is performed for the few elements with more than one crack.

An important aspect of the numerical application is the integration of the system matrices along Γ_{coh} of the cohesive cracks in Eq. (2.8). In many cases, a Gauss quadrature results in an overestimation of the stiffness, since the extreme values at the end of a crack segment might have passed the peak point in the interface law (small stiffness), whereas the Gauss point can still be within the penalty part of the interface law (high stiffness). In order to circumvent this problem, a Newton-Cotes integration with 3 integration points per segment has been applied, and a better convergence behavior of the numerical solution could be observed [46].



(a) geometry (b) with mapping (c) without mapping

Figure 2.11 Influence of the mapping procedure on the first principal stress for a hard inclusion in a soft matrix.

2.8.4 Solution of nonlinear system of equations

Most of the presented examples are calculated using a displacement control solution scheme, since a pure load-control is not able to simulate a response in the post-peak region. The standard displacement control approach fails, if within the simulation snap back phenomena occur. In this case, coupled schemes such as arc-length methods or indirect displacement control approaches have to be used. Within the solution procedure, the displacements corresponding to an equilibrium between the internal and external forces are to be determined.

From the principle of virtual displacements, it follows for a body in equilibrium:

$$\delta W_{int}^{(t+1)} = \delta W_{ext}^{(t+1)}, \tag{2.72}$$

where the variation of the internal energy $\delta W_{int}^{(t+1)}$ and the variation of the external energy $\delta W_{ext}^{(t+1)}$ is given by

$$\delta W_{int}^{(t+1)} = \int_V \boldsymbol{\sigma}^T(\boldsymbol{d}^{(t+1)})\delta\boldsymbol{\varepsilon}^{(t+1)}\, dV \tag{2.73}$$

$$= \int_V \left(\boldsymbol{\sigma}(\boldsymbol{d}^{(t)}) + \Delta\boldsymbol{\sigma}\left(\boldsymbol{d}^{(t)} + \Delta\boldsymbol{d}^{(t+1)}\right)\right)^T \delta\boldsymbol{\varepsilon}^{(t+1)}\, dV \tag{2.74}$$

$$\delta W_{ext}^{(t+1)} = \left(\boldsymbol{F}_{ext}^{(t)} + \Delta\boldsymbol{F}_{ext}^{(t+1)}\right)^T \delta\boldsymbol{d}^{(t+1)}. \tag{2.75}$$

Linearization of the system of equations gives

$$\int_V \left(\frac{\partial\sigma(\boldsymbol{d}^{(t)})}{\partial\varepsilon}\Delta\boldsymbol{\varepsilon}^{(t+1)}\right)^T \delta\boldsymbol{\varepsilon}^{(t+1)}\, dV = \left(\Delta\boldsymbol{F}_{ext}^{(t+1)}\right)^T \delta\boldsymbol{d}^{(t+1)} - \int_V \left(\boldsymbol{\sigma}^{(t)}\right)^T \delta\boldsymbol{\varepsilon}^{(t+1)}\, dV + \left(\boldsymbol{F}_{ext}^{(t)}\right)^T \delta\boldsymbol{d}^{(t+1)}, \tag{2.76}$$

where the last two summands on the right hand side vanish, if the previous step is in equilibrium. In the general formulation, the vector $\boldsymbol{d}$ includes all the nodal displacements. However,

if kinematic boundary conditions have to be applied, the variations of the displacement state have to be kinematically compatible. In general, the linear constraint equations can be written as

$$\boldsymbol{T}_1^{(t+1)}\boldsymbol{d}_1^{(t+1)} + \boldsymbol{T}_2^{(t+1)}\boldsymbol{d}_2^{(t+1)} = \bar{\boldsymbol{b}}^{(t+1)}, \tag{2.77}$$

where the nodal displacements are separated into a vector of independent, unknown DOF's $\boldsymbol{d}_1$ and a vector of dependent DOF's $\boldsymbol{d}_2$. For a simulation using the displacement control approach without additional coupling equations, the matrix $\boldsymbol{T}_1^{(t+1)}$ vanishes and the matrix $\boldsymbol{T}_2^{(t+1)}$ is equal to the identity matrix with $\bar{\boldsymbol{b}}^{(t+1)}$ being the prescribed displacements. In the case of periodic boundary conditions (e.g. $d_k + d_j = 0$) or in the case of coupling a 'hanging node' via displacement constraints (e.g. $d_l = 0.5d_j + 0.5d_k$), both transformation matrices do not vanish. These matrices depend on the load step indicated by t, since these conditions might change between successive load steps (e.g. $\bar{\boldsymbol{b}}$ is modified in each increment of a displacement controlled analysis). The assignment of a degree of freedom to either the dependent or the independent DOF's is sometimes arbitrary. For example in the case of the periodic boundary conditions, either d_k or d_j are independent, whereas the other is then the dependent DOF. Using a Gauss elimination procedure, which is equivalent to an inversion of the matrix $\boldsymbol{T}_2^{(t+1)}$, the vector of total displacements $\boldsymbol{d}$ can be expressed as a function of the independent DOF's $\boldsymbol{d}_1$ only:

$$\boldsymbol{d}^{(t+1)} = \begin{bmatrix} \boldsymbol{d}_1^{(t+1)} \\ \boldsymbol{d}_2^{(t+1)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} \\ -\boldsymbol{C}_{con} \end{bmatrix} \boldsymbol{d}_1^{(t+1)} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{b}^{(t+1)} \end{bmatrix}, \tag{2.78}$$

where it is assumed that the vector $\boldsymbol{d}^{(t+1)}$ is reordered so that the independent parameters are at the top. Note the difference between the vector $\bar{\boldsymbol{b}}^{(t+1)}$ in Eq. (2.77) and $\boldsymbol{b}^{(t+1)}$ in Eq. (2.78). The matrix $\boldsymbol{C}_{con}$ relates the dependent DOF's with the independent DOF's. In the further derivation it is assumed that the matrix C does not depend on the load step, since the increase of the loading is only obtained by modification of the right hand side $\boldsymbol{b}$.

Consequently, the variation of the nodal displacement vector is given by

$$\delta\boldsymbol{d}^{(t+1)} = \begin{bmatrix} \boldsymbol{I} \\ -\boldsymbol{C}_{con} \end{bmatrix} \delta\boldsymbol{d}_1^{(t+1)} \tag{2.79}$$

Using the element shape functions, the relation between nodal displacements $\boldsymbol{d}$ and strains $\boldsymbol{\varepsilon}$ can be expressed as

$$\Delta\boldsymbol{\varepsilon}^{(t+1)} = \boldsymbol{\varepsilon}^{(t+1)} - \boldsymbol{\varepsilon}^{(t)} \tag{2.80}$$

$$= \boldsymbol{B}\left(\begin{bmatrix} \boldsymbol{I} \\ -\boldsymbol{C}_{con} \end{bmatrix} \left(\boldsymbol{d}_1^{(t+1)} - \boldsymbol{d}_1^{(t)} \right) + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{b}^{(t+1)} \end{bmatrix} - \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{b}^{(t)} \end{bmatrix} \right) \tag{2.81}$$

$$\delta\boldsymbol{\varepsilon}^{(t+1)} = \boldsymbol{B}\left(\begin{bmatrix} \boldsymbol{I} \\ -\boldsymbol{C}_{con} \end{bmatrix} \delta\boldsymbol{d}_1^{(t+1)} \right), \tag{2.82}$$

where the relation from Eq. (2.78) has been applied. Setting

$$\Delta\boldsymbol{d}_1^{(t+1)} = \left(\boldsymbol{d}_1^{(t+1)} - \boldsymbol{d}_1^{(t)} \right) \tag{2.83}$$

and substituting Eqs. (2.79), (2.81) and (2.82) into Eq. (2.76) gives

$$\left(\begin{bmatrix}\boldsymbol{I}\\ -\boldsymbol{C}_{con}\end{bmatrix}\Delta\boldsymbol{d}_1^{(t+1)}+\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{b}^{(t+1)}\end{bmatrix}-\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{b}^{(t)}\end{bmatrix}\right)^T\boldsymbol{K}^T\left(\begin{bmatrix}\boldsymbol{I}\\ -\boldsymbol{C}_{con}\end{bmatrix}\delta\boldsymbol{d}_1^{(t+1)}\right)$$
$$=\left(\Delta\boldsymbol{F}_{ext}^{(t+1)}+\boldsymbol{F}_{ext}^{(t)}-\boldsymbol{F}_{int}^{(t)}\right)^T\begin{bmatrix}\boldsymbol{I}\\ -\boldsymbol{C}_{con}\end{bmatrix}\delta\boldsymbol{d}_1^{(t+1)}. \quad (2.84)$$

The integrated stiffness matrix and the internal and external force vectors are split into submatrices similar to $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$:

$$\int_V \boldsymbol{B}^T\frac{\partial\sigma(\boldsymbol{d}^{(t)})}{\partial\varepsilon}\boldsymbol{B}\,dV=\boldsymbol{K}=\begin{bmatrix}\boldsymbol{K}_{11}&\boldsymbol{K}_{12}\\ \boldsymbol{K}_{21}&\boldsymbol{K}_{22}\end{bmatrix} \quad (2.85)$$

$$\int_V \boldsymbol{B}^T\boldsymbol{\sigma}^{(t)}\,dV=\boldsymbol{F}_{int}^{(t)}=\begin{bmatrix}\boldsymbol{F}_{1,int}^{(t)}\\ \boldsymbol{F}_{2,int}^{(t)}\end{bmatrix}. \quad (2.86)$$

Canceling out $\delta\boldsymbol{d}_1^{(t+1)}$, transposing and splitting the external and internal forces into two parts yields:

$$\begin{bmatrix}\boldsymbol{I}&-\boldsymbol{C}_{con}^T\end{bmatrix}\begin{bmatrix}\boldsymbol{K}_{11}&\boldsymbol{K}_{12}\\ \boldsymbol{K}_{21}&\boldsymbol{K}_{22}\end{bmatrix}\left(\begin{bmatrix}\boldsymbol{I}\\ -\boldsymbol{C}_{con}\end{bmatrix}\Delta\boldsymbol{d}_1^{(t+1)}+\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{b}^{(t+1)}\end{bmatrix}-\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{b}^{(t)}\end{bmatrix}\right)$$
$$=\begin{bmatrix}\boldsymbol{I}&-\boldsymbol{C}_{con}^T\end{bmatrix}\begin{bmatrix}\Delta\boldsymbol{F}_{1,ext}^{(t+1)}+\boldsymbol{F}_{1,ext}^{(t)}-\boldsymbol{F}_{1,int}^{(t)}\\ \Delta\boldsymbol{F}_{2,ext}^{(t+1)}+\boldsymbol{F}_{2,ext}^{(t)}-\boldsymbol{F}_{2,int}^{(t)}\end{bmatrix}. \quad (2.87)$$

Expanding Eq. (2.87) finally gives:

$$\boldsymbol{K}^{mod}\Delta\boldsymbol{d}_1^{(t+1)}=\left(\boldsymbol{K}_{12}-\boldsymbol{C}_{con}^T\boldsymbol{K}_{22}\right)\left(\boldsymbol{b}^{(t)}-\boldsymbol{b}^{(t+1)}\right)+\Delta\boldsymbol{F}_{1,ext}^{(t+1)}+\boldsymbol{F}_{1,ext}^{(t)}$$
$$-\boldsymbol{F}_{1,int}^{(t)}-\boldsymbol{C}_{con}^T\left(\Delta\boldsymbol{F}_2^{(t+1)}+\boldsymbol{F}_{2,ext}^{(t)}-\boldsymbol{F}_{2,int}^{(t)}\right) \quad (2.88)$$

with the modified stiffness matrix

$$\boldsymbol{K}^{mod}=\boldsymbol{K}_{11}-\boldsymbol{C}_{con}^T\boldsymbol{K}_{21}-\boldsymbol{K}_{12}\boldsymbol{C}_{con}+\boldsymbol{C}_{con}^T\boldsymbol{K}_{22}\boldsymbol{C}_{con}. \quad (2.89)$$

Within the iteration procedure of a displacement controlled analysis, two special cases of Eq. (2.88) are important. The first case describes the modification of the boundary conditions, where it is assumed that the previous load step is in equilibrium. Consequently, Eq. (2.88) reduces to

$$\boldsymbol{K}^{mod}\Delta\boldsymbol{d}_1^{(t+1)}=\left(\boldsymbol{K}_{12}-\boldsymbol{C}_{con}^T\boldsymbol{K}_{22}\right)\left(\boldsymbol{b}^{(t)}-\boldsymbol{b}^{(t+1)}\right)+\Delta\boldsymbol{F}_{1,ext}^{(t+1)}-\boldsymbol{C}_{con}^T\Delta\boldsymbol{F}_2^{(t+1)}. \quad (2.90)$$

In the second special case, the boundary conditions and loads are not modified and the iteration is performed to obtain equilibrium. In this case, Eq. (2.88) simplifies to

$$\boldsymbol{K}^{mod}\Delta\boldsymbol{d}_1^{(t+1)}=\boldsymbol{F}_{1,ext}^{(t)}-\boldsymbol{F}_{1,int}^{(t)}-\boldsymbol{C}_{con}^T\left(\boldsymbol{F}_{2,ext}^{(t)}-\boldsymbol{F}_{2,int}^{(t)}\right). \quad (2.91)$$

The solution of the system of equations in a single iteration step is performed using the direct solver MUMPS [47].

The performance of the solution procedure can further be increased by an application of a line search procedure, which determines the length of a step given a search direction $\Delta \boldsymbol{d}^{(t+1)}$.

$$\boldsymbol{d}^{(t+1)} = \boldsymbol{d}^{(t)} + \eta \Delta \boldsymbol{d}^{(t+1)} \tag{2.92}$$

The requirement for a line search algorithm is that a sufficient decrease in the objective function has to be assured in order to guarantee convergence of the global solution. In this work, a line search based on the out-of-balance forces $\boldsymbol{F}_{oob}$ has been implemented, where the objective function is defined as

$$f(\boldsymbol{d}^{(t)}) = \left\| \boldsymbol{F}_{oob}(\boldsymbol{d}^{(t)}) \right\| \tag{2.93}$$

$$= \left\| \boldsymbol{F}^{(t)}_{1,ext} - \boldsymbol{F}^{(t)}_{1,int} - \boldsymbol{C}^T_{con} \left(\boldsymbol{F}^{(t)}_{2,ext} - \boldsymbol{F}^{(t)}_{2,int} \right) \right\| . \tag{2.94}$$

The line search is only performed within the iteration for an equilibrium state. It is based on the Goldstein-Armijo condition [48, 49, 50]:

$$\left\| \boldsymbol{F}_{oob}(\boldsymbol{d}^{(t)} + \eta \Delta \boldsymbol{d}^{(t+1)}) \right\| - \left| \boldsymbol{F}_{oob}(\boldsymbol{d}^{(t)}) \right| \leq \alpha \eta \nabla \left\| \boldsymbol{F}_{oob}(\boldsymbol{d}^{(t)}) \right\|^T \Delta \boldsymbol{d}^{(t+1)}. \tag{2.95}$$

with $\alpha \, \epsilon \, (0, 1)$. For a small value of α, the condition only requires a decrease in the objective function. For an optimization procedure with quadratic convergence (as the standard FEM without softening), a stronger requirement for the line search is preferable. Non-fulfillment of the criteria in Eq. (2.95) implies that the quadratic approximation of $f(\boldsymbol{d}^{(t)}$ is not valid at $\boldsymbol{d}^{(t)} + \Delta \boldsymbol{d}^{(t)}$, and, consequently, the load step was too large. Following [51], a value of $\alpha = 0.5$ has been used. From

$$\alpha \eta \nabla \left\| \boldsymbol{F}_{oob}(\boldsymbol{d}^{(t)}) \right\|^T \Delta \boldsymbol{d}^{(t+1)} = \alpha \eta \left(- \frac{\boldsymbol{K}^T \boldsymbol{F}_{oob}(\boldsymbol{d})}{|\boldsymbol{F}_{oob}(\boldsymbol{d})|} \right)^T \Delta \boldsymbol{d}^{(t+1)} \tag{2.96}$$

$$= \alpha \eta \left| \boldsymbol{F}_{oob}(\boldsymbol{d}) \right| , \tag{2.97}$$

the criteria in Eq. (2.95) can be rewritten as

$$\left\| \boldsymbol{F}_{oob}(\boldsymbol{d}^{(t)}) \right\| - \left\| \boldsymbol{F}_{oob}(\boldsymbol{d}^{(t)} + \eta \Delta \boldsymbol{d}^{(t+1)}) \right\| \geq \alpha \eta \left\| \boldsymbol{F}_{oob}(\boldsymbol{d}^{(t)}) \right\| . \tag{2.98}$$

At a time step t in the solution procedure, an increment $\Delta \boldsymbol{d}_1^{(t+1)}$ is calculated from Eq. (2.90). Starting with $\eta = 1$, the scaled increment $\eta \Delta \boldsymbol{d}_1^{(t+1)}$ is added to the displacements of the previous iteration $\boldsymbol{d}_1^{(t)}$ and the out-of-balance forces $\boldsymbol{F}_{oob}^{(t+1)}$ are determined. If the criteria in Eq. (2.98) is fulfilled, the lines search stops and the next iteration step using Eq. (2.91) is performed. Otherwise, the scale factor η is multiplied by 0.5 and the procedure is repeated. In the implementation, a minimum scale factor of 10^{-3} has been used. If the lower bound is reached, the linesearch stops and the displacement increment added to the previous step is scaled by 10^{-3}. Convergence in the overall Newton iteration of a load step is obtained, if the norm of the out-of-balance loads is smaller than a certain threshold. In the examples, a threshold between 10^{-4} and 10^{-6} has been used. The general solution procedure is summarized in Algorithm 2.2.

Algorithm 2.2 Nonlinear solution procedure

Initialize the structure and load increment
while (load < final load) **do**
  Increase load by load increment.
  Compute $\boldsymbol{d} = \boldsymbol{d} + \Delta\boldsymbol{d}$ using Eq. (2.90).
  Calculate out-of-balance loads $\boldsymbol{F}_{oob}(\boldsymbol{d})$.
  Initialize NumIterations = 0.
  while (NumIterations < 15) **and** ($F_{oob}(\boldsymbol{d}) >$ tol) **do**
    Compute $\Delta\boldsymbol{d}$ using Eq. (2.91).
    Initialize line search with $\eta = 1$
    while $\eta > 10^{-3}$ **and** $\dfrac{|\boldsymbol{F}_{oob}(\boldsymbol{d})| - |\boldsymbol{F}_{oob}(\boldsymbol{d} + \eta\Delta\boldsymbol{d})|}{|\boldsymbol{F}_{oob}(\boldsymbol{d})|} \leq \left(\dfrac{1}{2}\right)\eta$ **do**
      $\eta = 0.5\eta$.
    NumIterations = NumIterations + 1.
    $\boldsymbol{d} = \boldsymbol{d} + \eta\Delta\boldsymbol{d}$.
  if NumIterations < 15 **then**
    Update history variables.
  else
    Decrease load by load increment and multiply load increment by 0.5.

2.8.5 Parallelization

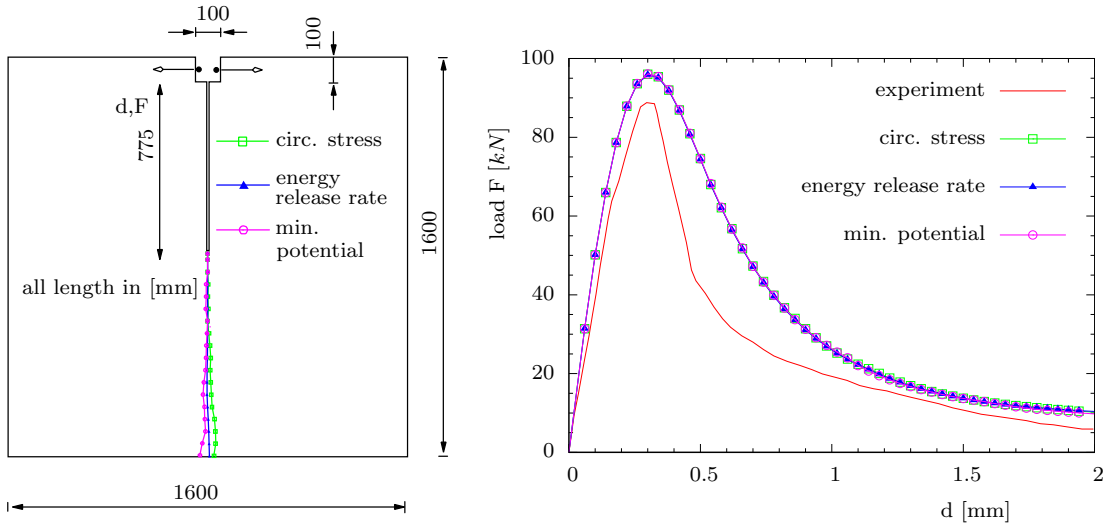
Simulations often require a fine resolution, which results in a large number of elements. In order to decrease the computation time, a parallelization of the algorithm is required. In this thesis, the focus of the parallelization techniques is placed on shared memory systems. The computational important tasks in the numerical procedure are the calculation of the residual force vector and the stiffness matrix and the solution of the system of equations. The latter is performed with solvers such as MUMPS [47], which directly include parallel versions. The parallel calculation of the system matrices can be performed in a straightforward way using OpenMP. The only interaction between the processors is required when assembling the local stiffness matrices into the global matrix.

In order to avoid that more than one processor modifies the same entry in the global stiffness matrix at the same time, an approach by [52] has been applied. This method decomposes the total set of elements into maximal independent sets. All elements in one set do not have a common entry in the global stiffness matrix. Consequently, the parallelization of the loop over all elements in one set can be efficiently implemented.

2.9 Examples

2.9.1 Wedge-splitting

In the first example, a wedge splitting test is investigated [53]. The geometry of the test specimen is illustrated in Fig. 2.12(a) with a specimen thickness of 400mm. Young's mod-



(a) geometry and crack path (b) load-displacement curve

Figure 2.12 Wedge-splitting test.

ulus, Poisson's ratio, uniaxial tensile strength and specific fracture energy are given as $E = 28300 N/mm^2$, $\nu = 0.18$, $f_{ct} = 2.11 N/mm^2$ and $G_f = 0.482 Nmm/mm^2$. In addition, the material parameters of the interface material law have been chosen as $K_p = 25\ 10^4 N/mm^3$ and $\alpha = 1$. In this mode I-dominated example, the choice of α has only a marginal influence on the result. The model consists of triangular elements with 6 nodes, but only the corner nodes are enriched with special purpose functions. Various criteria for the direction of a crack extension are compared.

It can be verified from Fig. 2.12(b) that the load-displacement curves obtained show a good correspondence with the experimental data without any parameter fitting applied. Although a slight variation of the crack path depending on the considered criteria for the direction of crack extension is obtained, their load-displacement curves are almost identical. The deviation from the theoretically straight vertical crack path is due to the non-symmetric mesh with triangular elements. A second source of error for the maximum energy dissipation criteria is the problem of partially closed crack extensions, which is discussed in detail in section 2.6.3.

2.9.2 L-shaped panel

In a second example, a mixed mode problem with a curved crack is investigated. The three presented approaches (maximum circumferential stress, maximum energy release rate and maximum of the dissipated energy) are compared in the numerical simulation with respect to the load-displacement curve and the crack path. The experiments for the L-shaped panel have been carried out by [53]. Figure 2.13 illustrates the geometry of the test specimen. The

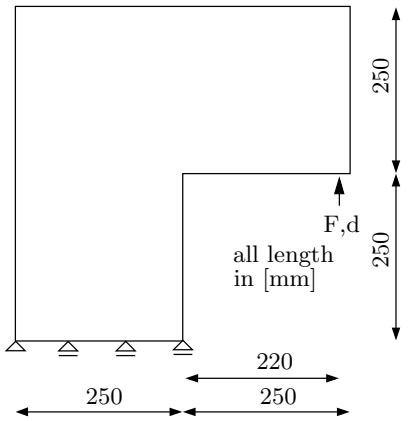


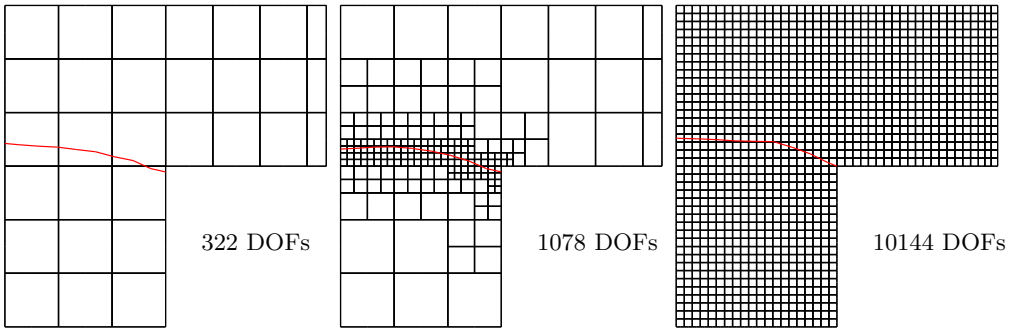
Figure 2.13 Experimental setup for the L-panel test.

thickness of the specimen is t = 100mm. A vertical displacement d is applied on the lower horizontal surface of the horizontal leg at a distance of 30mm from the vertical end face and the resulting vertical force at the same position is measured. Young's modulus, Poisson's ratio, uniaxial tensile strength and the specific fracture energy are given as $E = 25850 N/mm^2$, $\nu = 0.18$, $f_{ct} = 2.70 N/mm^2$ and $G_f = 0.09 Nmm/mm^2$. The material parameters have been adapted to match the initial elastic part, the peak load and the softening part of the load-displacement curve. The Young's modulus was modified to $E = 20000 N/mm^2$, the tensile strength $f_{ct} = 2.50 N/mm^2$ and the specific fracture energy $G_f = 0.13 Nmm/mm^2$. For the interface law, a penalty stiffness $K_p = 25\ 10^4 N/mm^3$ and a weighting factor $\alpha = 1$ has been used. A constant extension length l = 30mm of an evolving crack has been chosen.

The meshes applied are given in Fig. 2.14 using the quadtree refinement procedure. Quadrilateral elements with 9 nodes have been used, but only the corner nodes are enriched with additional special purpose functions. An example of the discretization obtained is illustrated in Fig. 2.14. Five different meshes are tested, a coarse and fine mesh with a constant element size in the domain and three meshes using the quadtree refinement procedure. Starting with an initially coarse mesh, the adaptation is performed at each crack extension. The mesh sensitivity of the model is tested by using different refinement levels of the quadtree structure, which means that an original element of the coarse mesh is refined into four subelements(adapt1), and these subelements are again refined into four subelements (adapt2) so that the width of an element is reduced by a factor of 0.5 at each adaptation step. As a result, the number of degrees of freedom (DOF's) increases from 322 DOF's for the initial (coarse) mesh to 1078 DOF's for the fully cracked model at the end of the simulation. In this example, the number of DOF's could be reduced by a factor of 10 compared to the fine mesh, which has the same element length as the adapted mesh in the vicinity of the crack.

The maximum tensile stress in the elastic part of the loading is obtained at an integration point close to the theoretical location. As a result, the initial crack does not start exactly at the corner. For the interpretation of the experimental results, the origins of the numerically obtained crack paths were moved to the inner corner of the L-panel.

For the criterion using the minimum potential energy, a total number of 15 directions within the range [-30^o,30^o] was tested. The internal energy for different extension angles for the



(a) coarse mesh (b) 3 adaptation steps (c) fine mesh

Figure 2.14 Regular quadrilateral element mesh obtained by using an adaptive quadtree refinement algorithm compared to the initial coarse mesh and the fine mesh.

first crack increment and the approximation using a moving least squares approximation is plotted in Fig. 2.15. The minimum of the internal energy is clearly visible at 5^o, which leads to a kinking of the second crack segment with respect to the initial crack.

The numerically-obtained crack paths and the corresponding load-displacement curves for different meshes are presented in Fig. 2.16. The maximum circumferential stress criterion already yields for the coarse mesh an accurate crack path, which is well in the experimental range. For the coarse mesh, a staircase trend is clearly visible in the load-displacement curve. This is due to the fact that the crack opens in an element-by-element way. Since the coarse mesh consist of only 3 elements cut by the crack, the opening of each element is recognizeable in the load-displacement curve. For finer meshes, this effect is reduced and, with an element length of 12.5mm (adapt2), there is almost no difference compared to the fine reference mesh.

Figure 2.16(b) shows the results for the criterion using the maximum energy release rate. A solution for the initial coarse mesh without any refinement could not be obtained; for the J-integral calculation, a certain number of elements around the crack tip are required to calculate the integral in Eq. (2.43). For this example, it can be observed that, compared to the maximum circumferential stress criterion, a finer mesh is required to converge to the final crack path. Furthermore, the obtained crack path shows a slightly higher curvature.

The results using the minimum potential energy criterion are illustrated in Fig. 2.16(c). The scatter of the obtained crack path is due to the constant extension length $\mathrm{l} = 30$mm and to the problems of partially closed elements discussed in section 2.6.3. For the coarse mesh, the element length was 83mm. Even for the first adaptation with an element length of 42mm, the extension length was too short compared to the element length. In general, it is recommended that the extension length is at least 2 to 3 times longer than the elements, which is confirmed in this numerical example. In addition to the effect of partially closed elements, a second reason for the small discrepancies between the fine mesh and the mesh with 3 quadtree adaptation steps is the biased adaptation strategy; this means that, for each crack direction, an individual adaptation of the mesh has been performed, resulting in different meshes for different directions. However, the crack path obtained with the finest mesh almost coincides with results obtained with the maximum circumferential stress criterion and the maximum

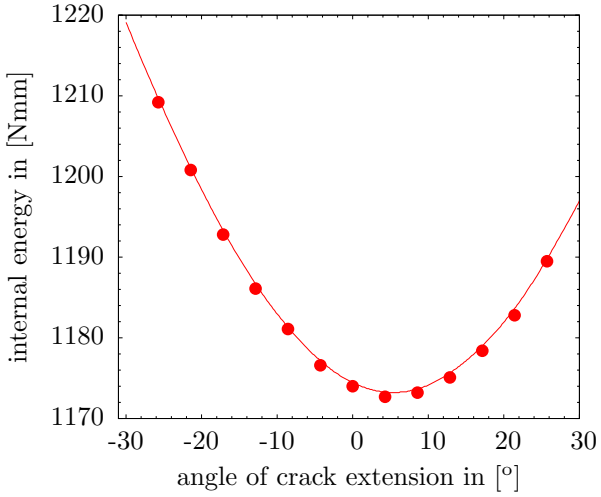


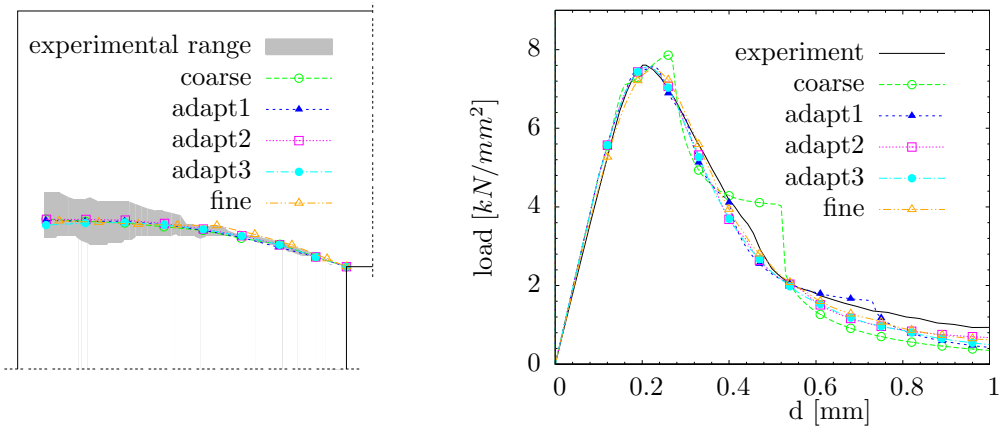
Figure 2.15 Internal energy of system for various angles of crack extension.

energy release rate; this indicates that the cohesive forces transferred through the crack (influencing the direction of crack extension only in the minimum potential energy criterion) have only a marginal influence in this case.

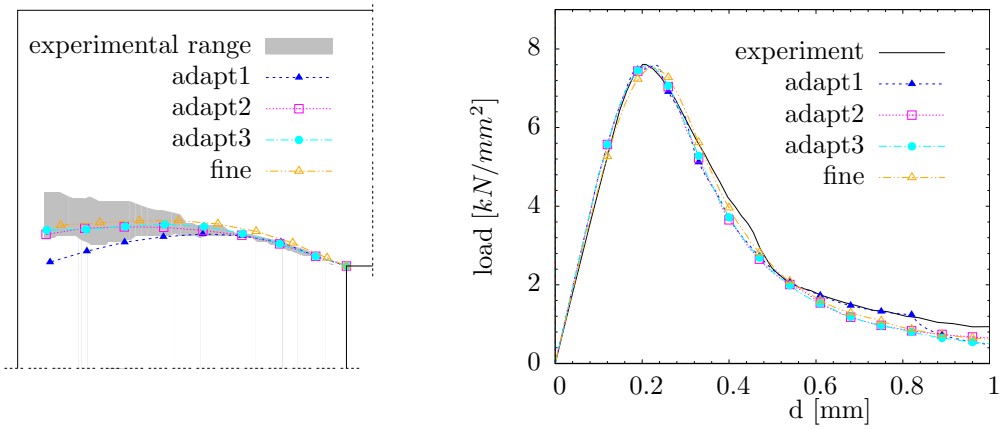
For this example, the popular nonlocal stress criterion [4] has been investigated in a preliminary investigation, in which the direction of the crack extension is assumed to be orthogonal to the eigenvector corresponding to the maximum principal nonlocal stress at the crack tip. Nonlocal stresses are used to improve the accuracy of the method, since the local stress state at the crack tip strongly depends on the discretization and, in the case of stress-free cracks, becomes infinite. The main problem is represented by the substantially different stress states parallel to the crack in the bulk material on both sides of the crack. In this example, the bulk material close to the upper crack face has much greater stresses in the horizontal direction than the bulk material in the lower part. Averaging these stresses does not yield an accurate prediction of the stress state at the crack tip, and a deviation from the correct crack path is obtained. In order to circumvent this problem, only points in front of the crack tip have been used to compute the nonlocal stresses, but the results could not be significantly improved.

2.9.3 Mixed mode fracture test

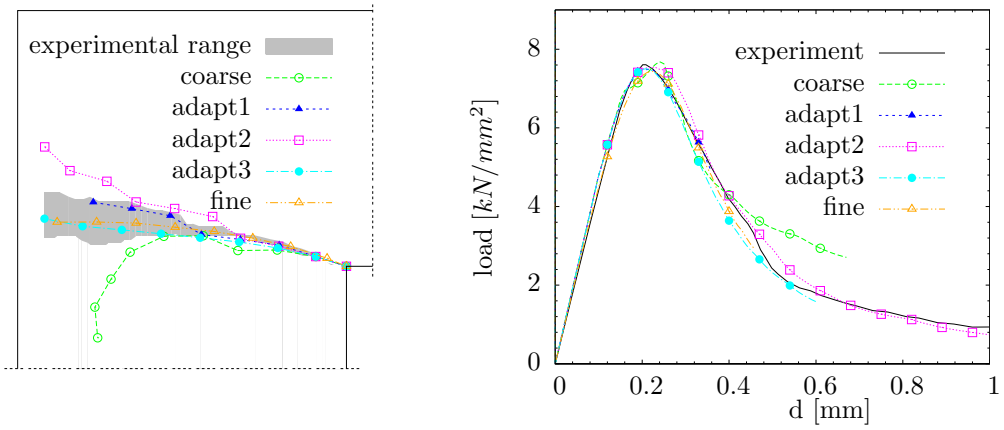
The final example is a mixed-mode fracture test performed by [54]. In this work, only the fracture tests 4a and 4b are considered (specimen 48-03 and 46-05). The dimensions are 200x200x50mm with a notch depth of 25mm and a notch width of 5mm, as illustrated in Fig. 2.17. The compressive strength from cubes with a dimension of 150mm and the splitting tensile strength are given as $f_c = 46.24\text{N/mm}^2$, $f_s = 3.67\text{N/mm}^2$ for specimen 4a and $f_c = 49.66\text{N/mm}^2$, $f_s = 3.76\text{N/mm}^2$ for specimen 4b. The Young's modulus, Poisson's ratio and fracture energy were not measured and, as with [55], they were estimated as $E = 30000\text{N/mm}^2$, $\nu = 0.2$ and $G_f = 0.110\text{Nmm/mm}^2$. The uniaxial tensile strength was estimated from the splitting tensile strength as $f_{ct} = 3.0\text{N/mm}^2$, and, by choosing the param-



(a) maximum circumferential stress



(b) maximum energy release rate



(c) minimum potential energy

Figure 2.16 Crack path and load-displacement curve for the L-shaped panel with constant crack extension l=30mm for various adaptation levels using the quadtree refinement.

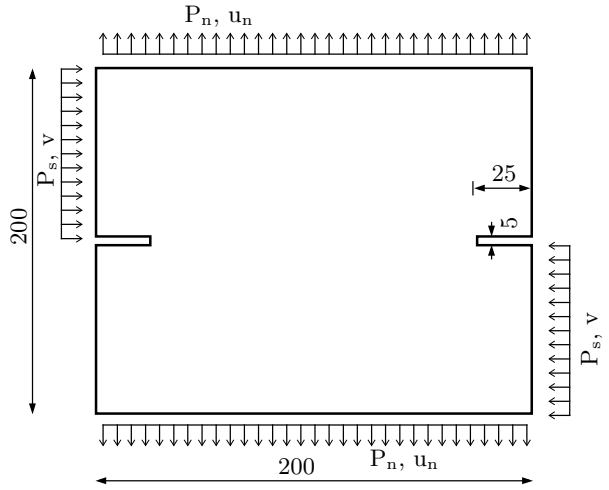


Figure 2.17 Geometry of specimen for the mixed mode fracture test.

eter α=0, the tangential cohesive forces were neglected. In a further study, the influence of the parameter α was investigated.

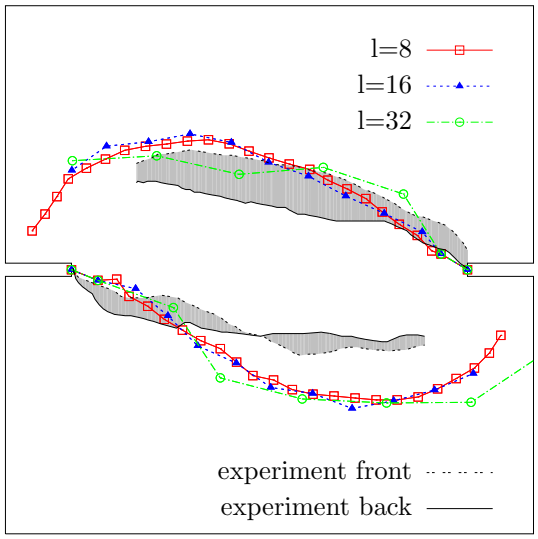
A shear force P_s was applied under displacement control up to 5kN for series 4a and 10kN for series 4b. The shear load was subsequently kept constant, and the specimen was loaded in normal direction P_n under displacement control. A constant mesh using triangular elements with an element length of 8mm was used.

For a shear load of 10kN the influence of the criterion for the direction of a crack extension and the influence of the material parameter α describing the influence of the shear stresses is investigated. The results are illustrated in Fig. 2.18.

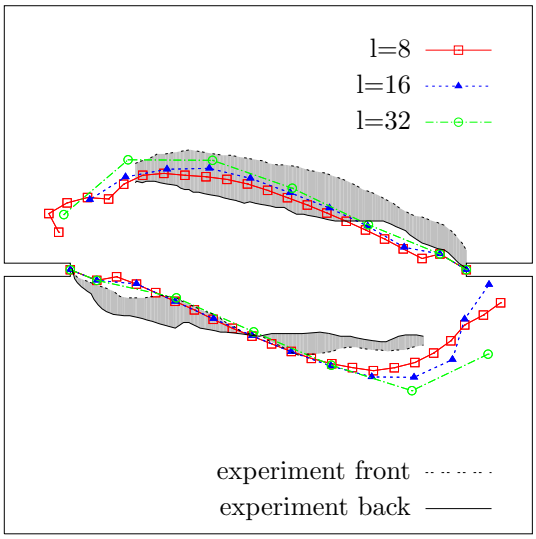
For the longest extension length $l = 32$mm and using the maximum circumferential stress criterion, the crack path in Fig. 2.18(a) showed oscillations around the exact crack path compared to results with a shorter extension length. This is due to an overestimation of the kink angle in the first crack extension step. However, almost no difference could be observed between the results for $l = 8$mm and $l = 16$mm.

A similar effect was found for the criterion using the maximum energy release rate in Fig. 2.18(b). It can also be noted that a long crack extension cannot capture regions of high curvature. As a result, the curvature of the crack path increases slightly with a shorter extension length.

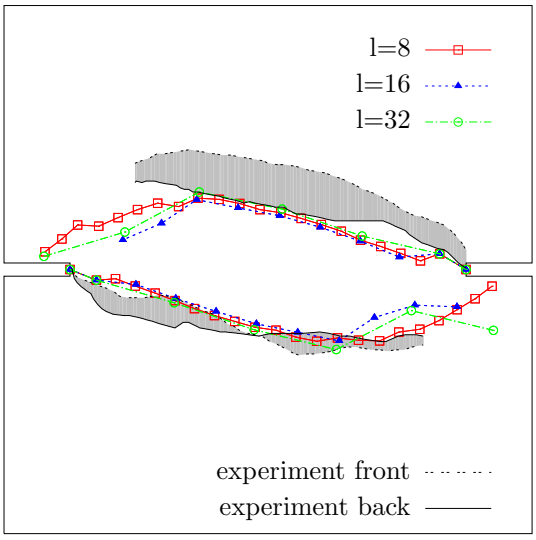
For the criterion using the minimum potential in Fig. 2.18(c), almost no influence of the extension length on the crack path is obtained. This is in contrast to the L-shaped panel. The reason is that, even for the shortest investigated extension length, the element length within the considered mesh is still small enough for more than one element to be influenced by a crack extension, and the mesh sensitivity is consequently reduced. Furthermore, the crack path obtained is almost straight, apart from two kinks in the upper and lower crack, which are present even in the coarse mesh. It can be concluded that, for mixed mode problems, the cohesive forces transferred through a crack have an influence on the crack direction (considered only in the minimum potential energy criterion) and can be neglected only for



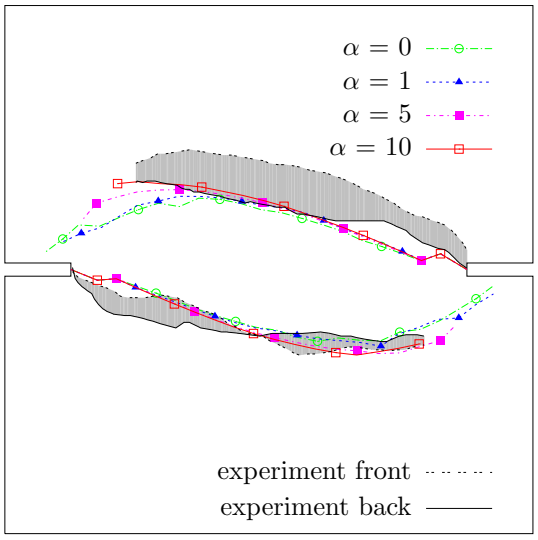
(a) Maximum circumferential stress



(b) Maximum energy release rate



(c) Minimum potential energy



(d) Influence of α using the minimum potential energy criterion and a crack extension of l=8mm

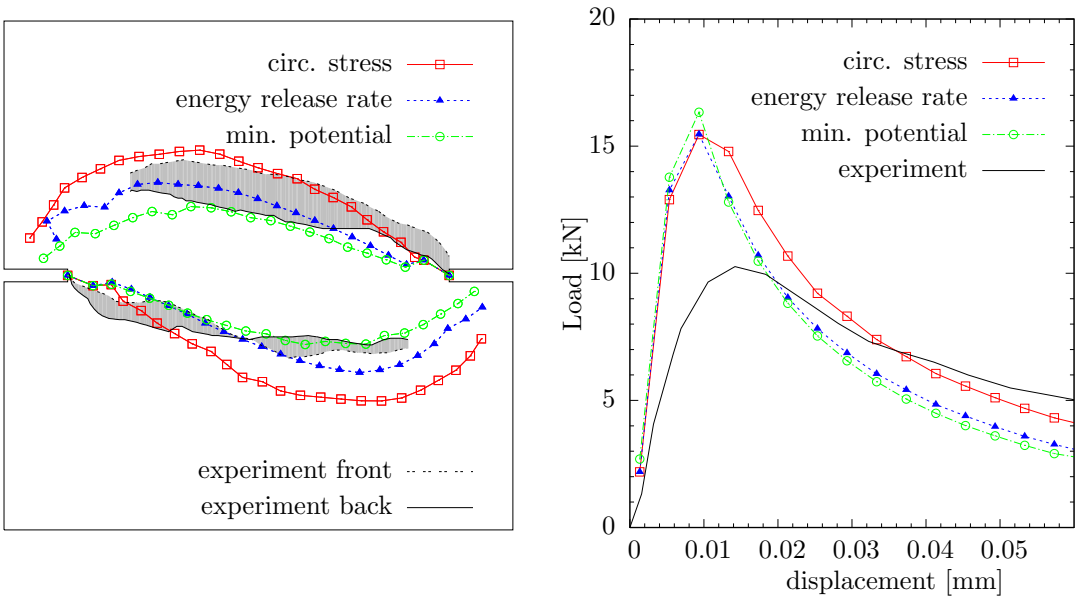
Figure 2.18 Influence of the crack extension length and the material parameter α on the obtained crack path using different direction criteria in the mixed mode facture test for a shear load of 10kN.

mode I-dominated problems. Additionally, the sensitivity of the crack length to crack paths with high curvatures is reduced. This is due to the fact that the criteria using the maximum energy release rate and the maximum circumferential stress are based on an explicit procedure calculating the angle of a crack extension using the last crack segment, whereas the criterion based on the minimum potential energy minimizes the potential energy of the system by adding a new segment in the corresponding direction.

Comparing the results for the various extension criteria as illustrated in Fig. 2.19(a), it can be observed that the maximum circumferential stress criterion yields the lowest curvature compared to the criterion using the maximum energy release rate; this is even lower than the curvature obtained with the algorithm that uses the minimum potential energy criterion. The differences in the crack paths have an influence on the load-displacement curves, as illustrated in Fig. 2.19(b). The energy dissipated in the crack is much higher for the maximum circumferential stress criterion due to the increased crack length. The initial stiffness and the maximum load from the numerical solution coincides with other results using the same material parameters with different methods [55]. Consequently, the difference between numerical solution and experimental results can probably be ascribed to the experimental setup or to the estimated material data. Compared to the experimental crack path, the calculated crack path for all three direction criteria investigated falls within the range of the experimental scatter.

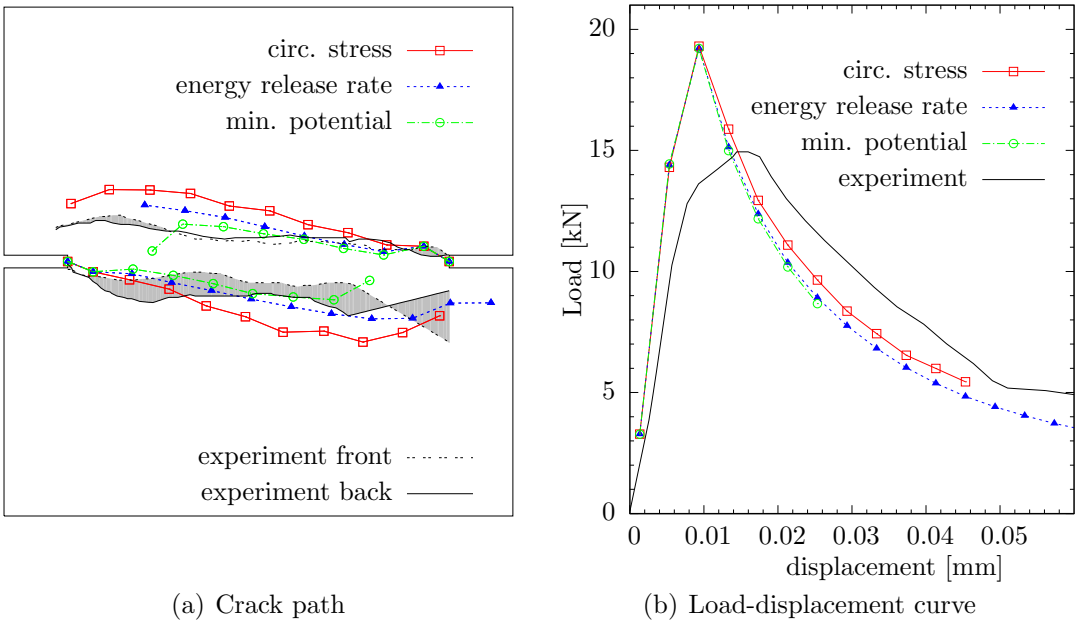
An interesting point is the determination of the parameter α, which determines the influence of the shear forces. For $\alpha = 0$, no tangential forces are considered, whereas for $\alpha = 1$, normal and tangential crack openings have an equal weight. From Fig. 2.18(d) it can be concluded that the sliding of the crack faces does not have a great deal of influence on the crack path as long as the latter remains almost straight. Only at the final stage, when the crack starts to curve, an influence of the cohesive shear forces can be observed. This can be attributed to the fact that the applied shear forces influence the stresses in front of the crack tip, and, consequently, the stress direction, whereas the tangential cohesive forces transferred through the cohesive interface are negligible as long as the tangential sliding of the crack faces remains minimal.

Similar results are obtained for a shear load of $P_s = 5$ kN given in Fig. 2.20(a). The highest curvature is obtained using the potential energy criterion, whereas the maximum circumferential stress criterion yields the lowest curvature. Due to the smaller shear force, a higher peak load in normal direction can be transmitted as illustrated in Fig. 2.20(b).



(a) Crack path (b) Load-displacement curve

Figure 2.19 Load-displacement curve for mixed mode fracture test with $P_s = 10$ kN and an extension length of l=8mm.



(a) Crack path (b) Load-displacement curve

Figure 2.20 Influence of the direction criteria for the crack extension in the mixed mode facture test for a shear load of 5kN and an extension length of 16mm.

Chapter 3

Continuum models for concrete

3.1 Introduction and state of the art

Fracture of concrete is characterized by the formation of microcracks in the fracture process zone, which coalesce into a macroscopic crack. The size of the fracture process zone is related to the microstructure of the material. At the formation of a macroscopic crack, localization takes place, which is often accompanied by a softening in the response. By using a continuum approach, this macroscopic crack is not considered as a discontinuity, but smeared across a certain width. In general, several approaches to model the nonlinear behavior of concrete can be distinguished.

A first approach is based on the plasticity formulation, where the total strain is decomposed into an elastic and a plastic part. The development of the plastic strain is governed by the yield function (e.g. for concrete [56, 57, 58]), the hardening rule, which describes the evolution of the yield surface, and the flow rule, which is related to a plastic potential that describes the incremental stress-strain relation. A comprehensive overview for plasticity models for concrete can be found in [59]. It is common practice to define the plasticity model in the stress space, although it can similarly be defined in the strain space [60]. The drawback of plasticity models is that the unloading stiffness remains elastic whereas in reality a degradation of the material with a reduction of the stiffness is observed.

This deficiency is overcome by models based on an approach introduced by Kachanov [61, 62], which was later termed continuum damage mechanics [63]. Damage in concrete is accompanied by the formation of microcracks. This phenomenon can be modeled by a fracture mechanics approach, which allows to describe the decohesion of existing crack faces. On the contrary, the continuum damage mechanics approach describes the effects of the microcracks such as stiffness degradation in a smeared sense. The simplest damage model is an isotropic damage model with a scalar variable ω that describes the damage state of the material - $\omega = 0$ corresponds to the virgin material, whereas for $\omega = 1$ the material is fully damaged and no stresses can be transfered. In [64], a comparison of the two approaches is given. A popular damage model for concrete was developed by [65], which defines different damage functions for compression and tension. Extending these ideas, anisotropic damage models [66, 67, 68] have been developed, which consider the direction of damage and are able to model dilatancy effects.

A third approach are the smeared crack models, which were introduced by Rashid [69]. The stress transfered through an element is directly related to the inelastic strain. Upon the development of inelastic strains, the initially isotropic material becomes orthotropic. Originally,

the direction of a crack in an element was assumed to remain constant, which lead to the so called fixed crack models [70]. Later, the multi-directional fixed smeared crack concept was introduced by Litton [71], where several cracks with a fixed orientation may interact. The problem of misalignment between principal axes of stress, strain and crack direction lead to the rotating crack concept [72, 73, 74]. A comparison of the different smeared crack approaches can be found in [75, 76].

In contrast to the models presented so far, which define a direct relation between strains and stresses, the microplane models based on [77], enhanced by [78] for metals and applied to concrete by [79, 80, 81, 82], define a stress-strain relation on various planes with different orientations. By integrating over all planes, the macroscopic constitutive relation is obtained.

In the following, an isotropic damage model is shortly reviewed and a model which couples damage and plasticity is presented.

3.2 Isotropic damage model

3.2.1 Local formulation

The isotropic damage model is based on the assumption that the stiffness degradation is isotropic and Poisson's ratio is not affected by the damage evolution. Consequently, a single scalar damage parameter ω is sufficient to describe the influence of damage on the stress-strain relationship by

$$\boldsymbol{\sigma} = (1-\omega)\boldsymbol{C}\boldsymbol{\varepsilon}, \tag{3.1}$$

with $\boldsymbol{\sigma}, \boldsymbol{\varepsilon}$ the stress and strain tensor and $\boldsymbol{C}$ the linear elastic material matrix. For the undamaged material $\omega = 0$. As the material is deformed, microcracks are initiated and propagate, and an overall decrease of stiffness is obtained. Consequently, the damage evolution can be described as a function of the total strain. In order to distinguish between loading and unloading, an equivalent strain κ is introduced that corresponds in 1D to the maximum strain ever reached in the loading history.

The scalar damage variable ω is a function of the equivalent strain κ. An exponential softening law is used [83]:

$$\omega(\kappa) = \begin{pmatrix} 0 & \kappa < \varepsilon_0 \\ 1.0 - \frac{\varepsilon_0}{\kappa} e^{\frac{\varepsilon_0 - \kappa}{\varepsilon_f - \varepsilon_0}} & \kappa \geq \varepsilon_0 \end{pmatrix}. \tag{3.2}$$

where ε_0 and ε_f are material constants.

The evolution of the parameter κ is described by a loading function f together with the Karush-Kuhn-Tucker conditions. The loading function is given by

$$f(\boldsymbol{\varepsilon}, \kappa) = \varepsilon^{eq}(\boldsymbol{\varepsilon}) - \kappa, \tag{3.3}$$

where the equivalent strain $\varepsilon^{eq}(\boldsymbol{\varepsilon})$ is expressed as

$$\varepsilon^{eq}(\boldsymbol{\varepsilon}) = \frac{1}{E}\sqrt{\sum_{I=1}^{3} \langle \sigma_I(\boldsymbol{\varepsilon}) \rangle^2}, \tag{3.4}$$

with $\langle \sigma_I \rangle$ the positive part of the principal stress and E the Young's modulus. Using this definition of the equivalent strain together with $\varepsilon_0 = \frac{f_{ct}}{E}$, the elastic domain is bounded by a rounded Rankine type failure surface.

The Karush-Kuhn-Tucker conditions

$$f \leq 0, \qquad \dot{\kappa} \geq 0, \qquad \dot{\kappa} f = 0, \tag{3.5}$$

ensure that the current equivalent strain is never larger than the equivalent strain κ, that the equivalent strain κ can only increase and that an increase of the equivalent strain κ can only be obtained for $\kappa = \varepsilon^{eq}(\boldsymbol{\varepsilon})$.

The specific fracture energy density g_f, which is the area under the uniaxial stress-strain curve, is given by

$$g_f = \int_{\varepsilon=0}^{\infty} \sigma \, d\varepsilon = \int_{\varepsilon=0}^{\varepsilon_0} E\varepsilon \, d\varepsilon + \int_{\varepsilon=\varepsilon_0}^{\infty} \varepsilon_0 e^{\frac{\varepsilon_0 - \kappa}{\varepsilon_f - \varepsilon_0}} E \, d\varepsilon \tag{3.6}$$

$$= \frac{E\varepsilon_0^2}{2} + \varepsilon_0 (\varepsilon_f - \varepsilon_0) E \tag{3.7}$$

$$= f_{ct} \left(\varepsilon_f - \frac{\varepsilon_0}{2} \right). \tag{3.8}$$

Consequently, the material parameter ε_f can be expressed as a function of the specific fracture energy density g_f

$$\varepsilon_f = \frac{g_f}{f_{ct}} + \frac{f_{ct}}{2E}. \tag{3.9}$$

In general, the specific fracture energy G_f is defined as the energy dissipated per crack surface, which can be obtained in an experimental setup as the quotient of the total dissipated energy and the crack surface.

In a finite element implementation with continuum elements, energy is not dissipated by separating the crack surfaces, but in a volume. Consider a one-dimensional beam with cross sectional area A under a tensile test. After failure, the total energy dissipated is given by $G_f A$, where G_f is an energy per area. Assuming that localization takes place in a single element with a constant energy dissipation density g_f (energy per volume), the total energy obtained by integration over the volume is $g_f A l_{eq}$, where l_{eq} is the length of the element. In order to avoid a strong influence of the mesh size on the results, the total dissipated energy for both approaches should be identical. As a result, the fracture energy density g_f used in the numerical implementation is given by:

$$g_f = \frac{G_f}{l_{eq}}. \tag{3.10}$$

In a three-dimensional element, the equivalent length l_{eq} depends on the orientation of the damage zone related to the element, which is difficult to follow in an iterative procedure. In the numerical implementation, the equivalent length is obtained by

$$l_{eq} = \begin{cases} \sqrt{A} & \text{plane stress} \\ \sqrt[3]{V} & \text{full three dimensional,} \end{cases} \tag{3.11}$$

where A and V are the area and the volume of the element. This assumption is exact for square and cubic elements with a constant edge length and a loading direction perpendicular to an edge.

3.2.2 Nonlocal formulation

The scaling procedure of the fracture energy using the equivalent length has several drawbacks. First of all, the zone of localization reduces to zero if the mesh is refined [84], thus the numerical solution is unobjective with respect to the mesh size. Furthermore, the scaling with a constant equivalent length is not correct - especially for triangular or distorted elements or in the case of an arbitrary loading direction.

In order to overcome these problems, a regularization technique based on the nonlocal formulation has been applied, where the stress state of a material point is not only a function of the variables at that specific point. This idea has been motivated by the fact that any material has a characteristic length at least on the atomistic level. If the resolution is smaller than this characteristic length, the assumptions of a continuum model no longer hold. Nonlocal models can be interpreted as homogenizing the material response within a certain region whose radius is related to the characteristic length of the material.

Nonlocal continuum models were first developed for elasticity [85, 86], for plasticity [87] , for strain softening materials [88, 89] and for concrete by [90]. An extensive overview about nonlocal models can be found in [91]. These nonlocal formulations serve as localization limiter, and the zone of energy dissipation remains finite. Another class of localization limiters, which is closely related to nonlocal models, are gradient models [92, 93, 94].

In a nonlocal formulation, different quantities can be averaged. First of all, it is required that the elastic response remains identical in the local and the nonlocal model. Consequently, the strain itself in Eq. (3.1) should not be averaged. Jirásek investigated different approaches [95] and showed that, for a pure tensile test, certain nonlocal formulations show locking phenomena. Promising results were obtained when averaging the energy release rate or the strain, from which the damage parameter is calculated.

In the implemented nonlocal formulation, a regularized form of Eq. (3.4) is obtained, where the equivalent strain is computed from the nonlocal strain by weighted spatial averaging:

$$\varepsilon^{eq}\left(\bar{\boldsymbol{\varepsilon}}\right) = \frac{1}{E}\sqrt{\sum_{I=1}^{3}\left\langle\sigma_I\left(\bar{\boldsymbol{\varepsilon}}\right)\right\rangle^2}, \tag{3.12}$$

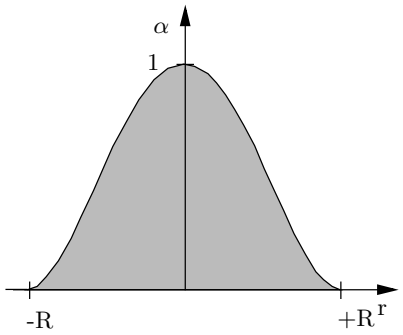


Figure 3.1 Nonlocal bell-shaped weight function.

where the nonlocal strain is calculated from

$$\bar{\boldsymbol{\varepsilon}}(\boldsymbol{x}) = \int_V w(\boldsymbol{x}, \boldsymbol{\xi}) \boldsymbol{\varepsilon}(\boldsymbol{x})\, d\boldsymbol{\xi}. \tag{3.13}$$

A bell-shaped weighting function illustrated in Fig. 3.1 has been used, which is rescaled in order to ensure that a constant local field induces a constant nonlocal field:

$$w(\boldsymbol{x}, \boldsymbol{\xi}) = \frac{\alpha(\|\boldsymbol{x} - \boldsymbol{\xi}\|)}{\int_V \alpha\left(\|\boldsymbol{x} - \boldsymbol{\eta}\|\right) d\boldsymbol{\eta}} \tag{3.14}$$

$$\alpha(r) = \left\{ \begin{array}{ll} \left(1 - \frac{r^2}{R^2}\right)^2 & r < R \\ 0 & r \geq R \end{array} \right\}. \tag{3.15}$$

In contrast to other weighting functions, the bell shaped weighting function has only a local support. This has the advantage that in the numerical implementation the bandwidth of the consistent stiffness matrix remains small compared to the matrix dimension. The influence radius R corresponds to the aforementioned length scale parameter. It should be chosen in the range of the intrinsic length of the material.

3.3 Plasticity model combined with nonlocal isotropic damage

3.3.1 Introduction

Pure plastic models are not capable of simulating the stiffness degradation, which is observed in experiments. By contrast, pure damage models are not able to represent irreversible deformations. A combination of both approaches can overcome these deficiencies. The combination of plasticity with damage is usually based on an isotropic plasticity model combined either with an anisotropic damage model as e.g. in [96, 97], or an isotropic damage model.

The plasticity model can either be formulated in the effective stress space (i.e. the undamaged stress space) as e.g. in [98, 99]. Another choice is a formulation in the nominal stress space (i.e. the damage stress space) as in [100, 101]. Grassl investigated the two different approaches with respect to the local uniqueness conditions [102], which means, whether for

a prescribed strain history a unique response in terms of stresses and history variables is obtained. The formulation in the effective stress space fulfills these conditions without any further restrictions, whereas for the formulation in the nominal stress space certain restrictions for the plasticity model (e.g. a plasticity model with hardening) have been found.

In the current work, a plasticity formulation in the effective stress space combined with an isotropic damage model similar to [103] has been used. An advantage from a numerical point of view is that the formulation in the effective stress space further allows a decoupling of the return mapping algorithm for the plasticity solution from the damage evolution.

The stress-strain relation is given by

$$\boldsymbol{\sigma} = (1-\omega)\,\boldsymbol{C} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^p) \tag{3.16}$$

$$= (1-\omega)\,\bar{\boldsymbol{\sigma}}, \tag{3.17}$$

where $\boldsymbol{\sigma}$ is the stress tensor, $\boldsymbol{C}$ the elastic material matrix, $\boldsymbol{\varepsilon}$ the total strain, $\boldsymbol{\varepsilon}^p$ the plastic strain, $\bar{\boldsymbol{\sigma}}$ the effective stress and

$$\bar{\boldsymbol{\sigma}} = \boldsymbol{C} : (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^p)\,. \tag{3.18}$$

For $\omega = 0$, the model corresponds to a plasticity model, whereas for $\boldsymbol{\varepsilon}^p = 0$ a pure damage model is obtained.

First, the plasticity model is presented, afterwards the local damage formulation is added, and, finally, the extension to the nonlocal damage model is made.

3.3.2 Plasticity formulation

An associated plasticity model defined in the effective stress space $\bar{\boldsymbol{\sigma}}$ is used. For multisurface plasticity, the evolution of the plastic strain is governed by Koiter's rule [104, 105]:

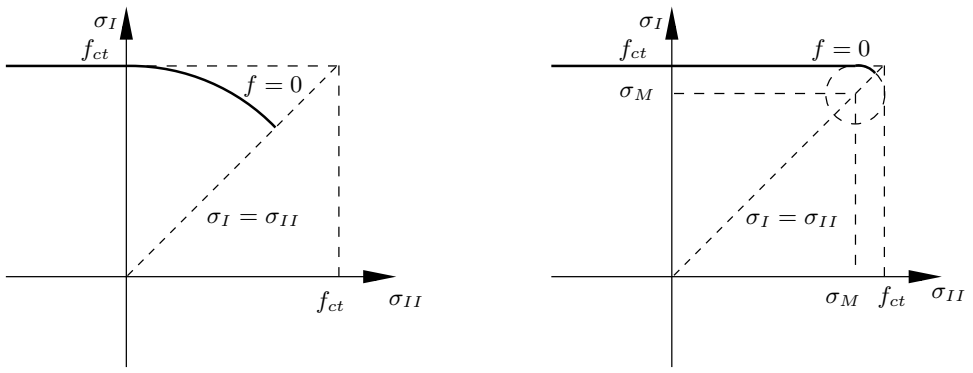
$$\dot{\boldsymbol{\varepsilon}}^p = \sum_{i=1}^{m} \dot{\gamma}_i \frac{\partial f_i^p}{\partial \bar{\boldsymbol{\sigma}}}, \tag{3.19}$$

where f_i^p are the yield functions of the plasticity model and γ_i the corresponding plastic multipliers. The Karush-Kuhn-Tucker conditions are given by

$$f_i^p \leq 0 \qquad \dot{\gamma}_i \geq 0 \qquad f_i^p \dot{\gamma}_i = 0 \qquad \forall i. \tag{3.20}$$

In this work, hardening is not considered in the model. Three different yield surfaces have been investigated - Rankine, Drucker-Prager and a combination of both within a multisurface plasticity approach. For the Rankine model, the yield surface is defined by

$$f_{RK}^p = \left\{ \begin{array}{ll} \sigma_I - f_{ct} & \sigma_{II,III} < 0 \\ \sqrt{\sigma_I^2 + \sigma_{II}^2} - f_{ct} & \sigma_{III} < 0 \\ \sqrt{\sigma_I^2 + \sigma_{II}^2 + \sigma_{III}^2} - f_{ct} & \sigma_{I,II,III} \geq 0 \end{array} \right\} \tag{3.21}$$



(a) rounded Rankine with radius $r = f_{ct}$

(b) rounded Rankine with radius $r < f_{ct}$

Figure 3.2 Rounded Rankine criterion for plane stress.

with the principal stresses $\sigma_I \geq \sigma_{II} \geq \sigma_{III}$ and the uniaxial tensile strength f_{ct}. The rounding of the yield surface is essential due to numerical reasons, since singularities occur in the standard Rankine approach for the derivative of the yield function at stress states with $\sigma_I = \sigma_{II}$. In the present approach, the rounding was performed with the full radius σ_y, but smaller radii can be used as illustrated in Fig. 3.2 to obtain in the limit with $\sigma_M \to f_{ct}$ the standard Rankine yield surface. This rounding corresponds to the assumption that the yield function is influenced by the interaction between positive principal stresses. The rounding with the third principal stress was only performed for the full 3-D simulation, whereas for the axisymmetric calculation, the rounding was only applied to the two principal stresses in the plane, since the hoop stress $\sigma_{\theta\theta}$ can directly be calculated (shear stresses $\sigma_{r\theta} = \sigma_{z\theta} = 0$). The yield surface (in combination with the Drucker-Prager yield surface) is illustrated in Fig. 3.4. For details on the calculation of the principal stresses in 3-D and its derivatives, see section B.1.

The Drucker-Prager yield surface is defined by

$$f_{DP}^p = \frac{\beta}{3} I_1 + \sqrt{J_2} - H_p \tag{3.22}$$

$$I_1 = \sigma_{xx} + \sigma_{yy} + \sigma_{zz} \tag{3.23}$$

$$J_2 = \frac{(\sigma_{xx} - \sigma_{yy})^2 + (\sigma_{xx} - \sigma_{zz})^2 - (\sigma_{yy} - \sigma_{zz})^2}{6} + \sigma_{xy}^2 + \sigma_{xz}^2 + \sigma_{yz}^2, \tag{3.24}$$

where I_1 is the first invariant of the stress tensor and J_2 the second invariant of its deviator. The material parameters β and H_p can be determined from the uniaxial compressive strength f_c and the biaxial compressive strength f_{c2}:

$$\beta = \frac{\sqrt{3}(f_{c2} - f_c)}{2f_{c2} - f_c} \tag{3.25}$$

$$H_p = \frac{f_{c2} f_c}{\sqrt{3}(2f_{c2} - f_c)}. \tag{3.26}$$

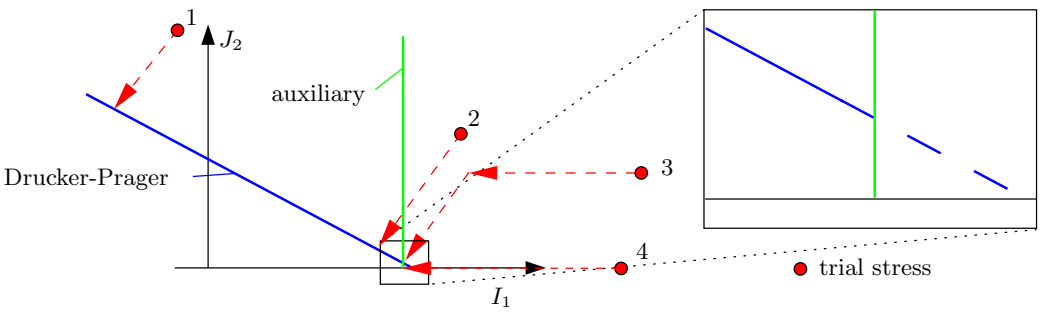


Figure 3.3 Return mapping of trial stresses using the Drucker-Prager yield surface in combination with the auxiliary yield function.

The Drucker-Prager yield surface corresponds to a cone in the principal stress space and has a singularity at the apex. In order to increase the stability and the convergence speed of the numerical solution close to this singularity, an auxiliary yield function has been added:

$$f^p_{aux} = \frac{\beta}{3} I_1 - \alpha H_p, \tag{3.27}$$

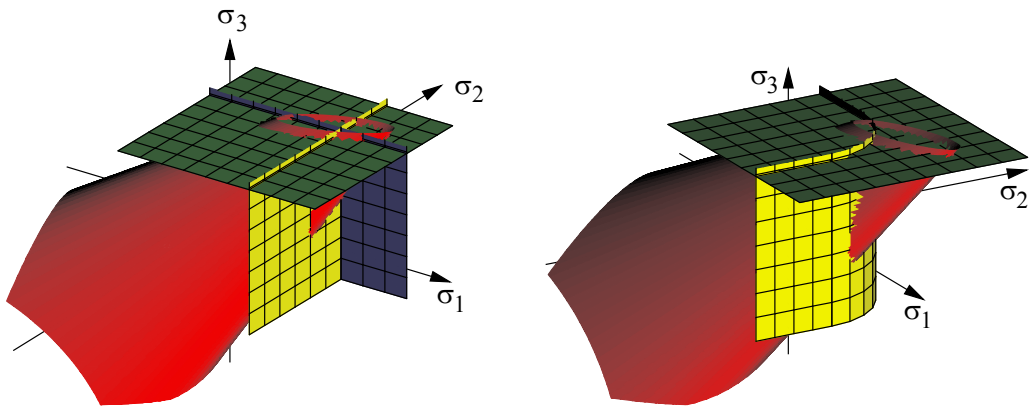
with $\alpha = 0.9999$. This is illustrated in Fig. 3.3. Trial stress 1 is directly mapped to the Drucker-Prager yield surface. For trial stress 2, the auxiliary yield function is activated for the first iteration step of the return mapping, but finally deactivated. Without the auxiliary function, numerical problems occur for trial stresses 3 and 4, but with the auxiliary yield function activated, they are both mapped back to the apex.

The Drucker-Prager yield surface gives a good approximation of the material behavior of concrete in compression. Other material laws for concrete in compression additionally include the Lode angle (or equivalently, the third invariant of the stress tensor), which has not been considered in this work. One of the main purposes of mesoscale simulations is the assumption that simple material formulations for each component lead to a complex response due to the geometrical distribution and the interaction of the components.

The Rankine criterion is a good approximation in tension. Consequently, a combination of both yield criteria to model tension as well as compression has been implemented, as illustrated in Figure 3.4. By using the rounded Rankine criterion no auxiliary yield function is required.

3.3.2.1 Numerical implementation

Due to the decoupling of plasticity and damage, a return mapping algorithm for the plastic part of the model with multisurface plasticity according to [106] can be used. Application



(a) Rankine and Drucker-Prager yield surfaces

(b) rounded Rankine and Drucker-Prager yield surfaces

Figure 3.4 Yield criteria for axisymmetric calculation.

of an implicit backward Euler difference scheme to Eqs. (3.19), (3.20) and (3.18) gives the following coupled system of equations:

$$\boldsymbol{\sigma}_{n+1} = \boldsymbol{C}\left(\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}^p_{n+1}\right) \tag{3.28}$$

$$\boldsymbol{\varepsilon}^p_{n+1} = \boldsymbol{\varepsilon}^p_n + \sum_{\alpha=1}^{m} \Delta\gamma_\alpha \frac{\partial f^p_\alpha(\boldsymbol{\sigma}_{n+1})}{\partial \boldsymbol{\sigma}} \tag{3.29}$$

$$f^p_\alpha(\boldsymbol{\sigma}_{n+1}) \leq 0 \tag{3.30}$$

$$\Delta\gamma_\alpha \geq 0 \tag{3.31}$$

$$\Delta\gamma_\alpha f^p_\alpha(\boldsymbol{\sigma}_{n+1}) = 0, \tag{3.32}$$

where $\boldsymbol{\sigma}$ is the stress tensor. In the current model with combined damage and plasticity, this variable corresponds to the effective stress tensor $\bar{\boldsymbol{\sigma}}$, but for comparability with classical plasticity formulations, the notation has been modified within this section.

If the trial stress with $\boldsymbol{\varepsilon}^p_{n+1} = \boldsymbol{\varepsilon}^p_n$ is in the elastic part of a single yield surface with $f^p_\alpha(\boldsymbol{\sigma}_{n+1}) < 0$ ($\boldsymbol{\sigma}_{n+1}$ is initialized with $\boldsymbol{\sigma}_{trial}$), it follows directly that $\Delta\gamma_\alpha = 0$. Consequently, only the active yield surfaces $\mathcal{J}$ with $f^p_\alpha(\boldsymbol{\sigma}_{n+1}) \geq 0$ are considered in the following, for which $\Delta\gamma_\alpha \neq 0$ and, at the end of the return mapping, $f^p_\alpha(\boldsymbol{\sigma}_{n+1}) = 0$.

Solving Eq. (3.28) for $\boldsymbol{\varepsilon}^p_{n+1}$ and substituting the result into Eq. (3.29) gives:

$$\boldsymbol{0} = \boldsymbol{C}^{-1}\boldsymbol{\sigma}_{n+1} - \boldsymbol{\varepsilon}_{n+1} + \boldsymbol{\varepsilon}^p_n + \sum_{\alpha: f_\alpha \epsilon \mathcal{J}} \Delta\gamma_\alpha \frac{\partial f^p_\alpha(\boldsymbol{\sigma}_{n+1})}{\partial \boldsymbol{\sigma}} \tag{3.33}$$

$$0 = f^p_\alpha(\boldsymbol{\sigma}_{n+1}) \quad \forall\alpha. \tag{3.34}$$

The only unknown variables within this system of equations are the stresses $\boldsymbol{\sigma}_{n+1}$ and the plastic multipliers $\Delta\gamma_\alpha$ of the active yield functions. In general, the trial state does not

fulfill the system of equations and residuals $\boldsymbol{R}^{(t)}$ and $f_\alpha^{(t)}$ are present. Within a local Newton iteration, a Taylor series expansion with respect to the unknowns $\boldsymbol{\sigma}_{n+1}$ and γ_α is performed:

$$\boldsymbol{R}^{(t+1)} = \boldsymbol{R}^{(t)} + \left[\boldsymbol{C}^{-1} + \sum_{\alpha:f_\alpha \epsilon \mathcal{J}} \Delta\gamma_\alpha \frac{\partial^2 f_\alpha^p(\boldsymbol{\sigma}_{n+1})}{\partial \boldsymbol{\sigma}^2}\right] \Delta\boldsymbol{\sigma}_{n+1} + \sum_{\alpha:f_\alpha \epsilon \mathcal{J}} \Delta\Delta\gamma_\alpha \frac{\partial f_\alpha^p(\boldsymbol{\sigma}_{n+1})}{\partial \boldsymbol{\sigma}} \tag{3.35}$$

$$\boldsymbol{f}^p(t+1) = \boldsymbol{f}^{p(t)} + \partial_\sigma \boldsymbol{f}^p(\boldsymbol{\sigma}_{n+1}) \Delta\boldsymbol{\sigma}_{n+1}, \tag{3.36}$$

where the equations that enforce the stress state to lie on the yield surface are combined into a single equation using the vector/matrix notation:

$$\Delta\boldsymbol{\sigma}_{n+1} = \begin{bmatrix}\Delta\sigma_{xx} & \Delta\sigma_{yy} & \Delta\sigma_{zz} & \Delta\sigma_{xy} & \Delta\sigma_{yz} & \Delta\sigma_{zx}\end{bmatrix}^T \tag{3.37}$$

$$\partial_\sigma \boldsymbol{f}^p(\boldsymbol{\sigma}_{n+1}) = \begin{bmatrix} \dfrac{\partial f_1^p(\boldsymbol{\sigma}_{n+1})}{\partial \sigma_{xx}} & \cdots & \dfrac{\partial f_1^p(\boldsymbol{\sigma}_{n+1})}{\partial \sigma_{zx}} \\ \vdots & \ddots & \\ \dfrac{\partial f_m^p(\boldsymbol{\sigma}_{n+1})}{\partial \sigma_{xx}} & & \dfrac{\partial f_m^p(\boldsymbol{\sigma}_{n+1})}{\partial \sigma_{zx}}. \end{bmatrix} \tag{3.38}$$

Solving Eq. (3.35) for $\Delta\boldsymbol{\sigma}_{n+1}$ and requiring the residuals at $(t+1)$ to vanish gives:

$$\Delta\boldsymbol{\sigma}_{n+1} = -\boldsymbol{\Sigma}\left(\boldsymbol{R}^{(t)} + \left[\partial_\sigma \boldsymbol{f}^p(\boldsymbol{\sigma}_{n+1})\right]^T \begin{bmatrix}\Delta\Delta\gamma_1 \\ \vdots \\ \Delta\Delta\gamma_m\end{bmatrix}\right), \tag{3.39}$$

where the abbreviation

$$\boldsymbol{\Sigma} = \left[\boldsymbol{C}^{-1} + \sum_{\alpha:f_\alpha \epsilon \mathcal{J}} \Delta\gamma_\alpha \frac{\partial^2 f_\alpha^p(\boldsymbol{\sigma}_{n+1})}{\partial \boldsymbol{\sigma}^2}\right]^{-1} \tag{3.40}$$

has been used. Substitution of Eq. (3.39) into Eq. (3.36) finally gives a coupled set of equations for the unknown plastic multipliers $\Delta\Delta\gamma_\alpha$:

$$\begin{bmatrix}\Delta\Delta\gamma_1 \\ \vdots \\ \Delta\Delta\gamma_m\end{bmatrix} = \left\{\partial_\sigma \boldsymbol{f}^p(\boldsymbol{\sigma}_{n+1}) \boldsymbol{\Sigma} \left[\partial_\sigma \boldsymbol{f}^p(\boldsymbol{\sigma}_{n+1})\right]^T\right\}^{-1} \left\{\boldsymbol{f}^{p(t)}(\boldsymbol{\sigma}_{n+1}) - \partial_\sigma \boldsymbol{f}^p(\boldsymbol{\sigma}_{n+1}) \boldsymbol{\Sigma} \boldsymbol{R}^{(t)}\right\}. \tag{3.41}$$

Starting from the plastic multipliers, back substitution gives from Eq. (3.29) the plastic strains and from Eq. (3.28) the stress at iteration $(t+1)$. The Newton iteration is repeated until, i.e. the Karush-Kuhn-Tucker conditions are fulfilled up to a certain accuracy. The algorithm in combination with the damage formulation is summarized in Algorithm 3.1 at the end of the section on page 55.

Note that the choice of the active yield functions does not remain constant during the iteration. For each intermediate state in the Newton iteration, the Karush-Kuhn-Tucker conditions are

verified. If the plastic multiplier for a formerly active yield function is negative, the yield function is removed from the set of active functions. Similarly, if the value of a yield function that is not part of the active set is positive, the function is added to the active set. In general, this procedure converges. However for certain situations, an oscillating behavior was experienced, where a yield function was added in one step and removed in the next one. In such a situation, where convergence after a certain number of iterations is not achieved, all possible sets of active yield functions are checked. Within each of these iterations, the set of active functions remains constant. After convergence of the procedure (all active yield functions are equal to zero), consistency of the solution is verified, i.e. if all other yield functions/plastic multiplier fulfill the Karush-Kuhn-Tucker conditions. In the rare case, where even this procedure did not result in a converged solution, the total strain increment was applied in steps. This approach has the drawback that the consistent tangent stiffness matrix is not exact, since the step length has an influence as shown in the next section. However, the line search procedure in combination with a step length control in the global Newton iteration detects an inexact stiffness matrix, and automatically reduces the global load step. In reality, the subdivision of the strain increment on the integration point level was only rarely required.

The nonlinear iteration on the system level is solved using a full Newton-Raphson iteration, which requires the determination of an algorithmic elasto-plastic tangent modulus. Differentiation of Eqs. (3.28) and (3.29) gives:

$$d\boldsymbol{\sigma}_{n+1} = \boldsymbol{C}\left(d\boldsymbol{\varepsilon}_{n+1} - d\boldsymbol{\varepsilon}^p_{n+1}\right) \tag{3.42}$$

$$d\boldsymbol{\varepsilon}^p_{n+1} = \sum_{\alpha: f_\alpha \epsilon \mathcal{J}} \left(\Delta\gamma_\alpha \frac{\partial^2 f^p_\alpha(\boldsymbol{\sigma}_{n+1})}{\partial \boldsymbol{\sigma}^2} d\boldsymbol{\sigma}_{n+1} \right) + \left[\partial_\sigma \boldsymbol{f}^p(\boldsymbol{\sigma}_{n+1})\right]^T d\Delta\boldsymbol{\gamma}. \tag{3.43}$$

Solving Eq. (3.42) for $d\boldsymbol{\varepsilon}^p_{n+1}$ and substituting the result into Eq. (3.43) gives

$$d\boldsymbol{\sigma}_{n+1} = \boldsymbol{\Sigma}\left[d\boldsymbol{\varepsilon}_{n+1} - \left[\partial_\sigma \boldsymbol{f}^p(\boldsymbol{\sigma}_{n+1})\right]^T d\Delta\boldsymbol{\gamma}\right] \tag{3.44}$$

where $\boldsymbol{\Sigma}$ is given by Eq. (3.40). Differentiation of the Karush-Kuhn-Tucker condition for the active yield functions and substitution into Eq. (3.42) gives an expression for $d\Delta\boldsymbol{\gamma}$:

$$0 = \partial_\sigma \boldsymbol{f}^p(\boldsymbol{\sigma}_{n+1})\, d\boldsymbol{\sigma}_{n+1} \tag{3.45}$$

$$= \partial_\sigma \boldsymbol{f}^p(\boldsymbol{\sigma}_{n+1})\, \boldsymbol{\Sigma}\left[d\boldsymbol{\varepsilon}_{n+1} - \left[\partial_\sigma \boldsymbol{f}^p(\boldsymbol{\sigma}_{n+1})\right]^T d\Delta\boldsymbol{\gamma}\right] \tag{3.46}$$

$$d\Delta\boldsymbol{\gamma} = \left\{\partial_\sigma \boldsymbol{f}^p(\boldsymbol{\sigma}_{n+1})\boldsymbol{\Sigma}\left[\partial_\sigma \boldsymbol{f}^p(\boldsymbol{\sigma}_{n+1})\right]^T\right\}^{-1} \left\{\partial_\sigma \boldsymbol{f}^p(\boldsymbol{\sigma}_{n+1})\boldsymbol{\Sigma} d\boldsymbol{\varepsilon}_{n+1}\right\}. \tag{3.47}$$

Note that the matrix $\left\{\partial_\sigma \boldsymbol{f}^p(\boldsymbol{\sigma}_{n+1})\boldsymbol{\Sigma}\left[\partial_\sigma \boldsymbol{f}^p(\boldsymbol{\sigma}_{n+1})\right]^T\right\}^{-1}$ has already been determined in Eq. (3.41). Substituting the expression from Eq. (3.47) and into Eq. (3.44) and rearranging, finally gives the expression for the algorithmic elasto-plastic tangent

$$\frac{d\boldsymbol{\sigma}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} = \boldsymbol{\Sigma} - \boldsymbol{\Sigma}\left[\partial_\sigma \boldsymbol{f}^p(\boldsymbol{\sigma}_{n+1})\right]^T \left\{\partial_\sigma \boldsymbol{f}^p(\boldsymbol{\sigma}_{n+1})\boldsymbol{\Sigma}\left[\partial_\sigma \boldsymbol{f}^p(\boldsymbol{\sigma}_{n+1})\right]^T\right\}^{-1} \partial_\sigma \boldsymbol{f}^p(\boldsymbol{\sigma}_{n+1})\boldsymbol{\Sigma}. \tag{3.48}$$

It is further to be noted that $\boldsymbol{\Sigma}$ is symmetric, which simplifies the computational effort. In combination with the damage formulation, the algorithmic tangent of the plastic strains with respect to the total strains is required. This can be derived from differentiation of Eq. (3.28):

$$\frac{d\boldsymbol{\sigma}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} = \boldsymbol{C} - \boldsymbol{C}\frac{d\boldsymbol{\varepsilon}^p_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}}. \tag{3.49}$$

Rearranging Eq. (3.49) gives the required expression:

$$\frac{d\boldsymbol{\varepsilon}^p_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}} = \boldsymbol{C}^{-1}\left(\boldsymbol{C} - \frac{d\boldsymbol{\sigma}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}}\right), \tag{3.50}$$

where $\dfrac{d\boldsymbol{\sigma}_{n+1}}{d\boldsymbol{\varepsilon}_{n+1}}$ can be calculated from Eq. (3.48). Note again that for a combined damage/plasticity formulation $\boldsymbol{\sigma}$ has to be replaced by the effective stress $\bar{\boldsymbol{\sigma}}$.

3.3.3 Local damage formulation

The proposed material law should be used within a mesoscale approach to model the soft matrix, which encloses the hard aggregates. For this scale, it is sufficient to simulate damage with an isotropic damage model, since the anisotropy of the global response is implicitly comprised in the mesoscale model. Several choices for the description of damage are possible, i.e. as a function of the stresses, the total strains or the plastic strains. In the presented model, the definition of a damage model as a function of the equivalent plastic strains has been used. Plasticity and damage start at the same time, and the elastic region is fully described by the yield function of the plasticity model. This corresponds to the assumption that relocations in the microstructure at the elastic limit are described by the plasticity model, but simultaneously lead to damage within the microstructure. With this approach, hardening in the pre-peak region of the load-displacement curve cannot be captured. However, this phenomena is included in the mesomodel due to the successive creation of microcracks, which coalesce only in the post-peak region into macroscopic cracks. The damage model is described by the evolution law, the loading function and the loading/unloading conditions. The loading function is given by

$$f^d(\boldsymbol{\varepsilon}^p) = \|\boldsymbol{\varepsilon}^p\| - \kappa, \tag{3.51}$$

with the loading/unloading conditions

$$f^d \leq 0 \qquad \dot{\kappa} \geq 0 \qquad f^d\dot{\kappa} = 0, \tag{3.52}$$

and the definition of the equivalent plastic strain

$$\|\boldsymbol{\varepsilon}^p\| = \sqrt{{\varepsilon^p_{xx}}^2 + {\varepsilon^p_{yy}}^2 + {\varepsilon^p_{zz}}^2 + 2\left({\varepsilon^p_{xy}}^2 + {\varepsilon^p_{xz}}^2 + {\varepsilon^p_{yz}}^2\right)}. \tag{3.53}$$

The damage evolution is given by

$$\omega = 1 - e^{-\frac{\kappa}{\varepsilon_f}}. \tag{3.54}$$

The parameter ε_f is a parameter that is related to the fracture energy G of the material and the equivalent length l_{eq}, which is a parameter of the finite element model and describes the influence radius of the corresponding material point, i.e. the dimension of the element that includes this material point. Similar to section 3.2.1, the parameter can be determined from the area under the stress-strain curve. Assume a one-dimensional tensile test simulated with the combined plasticity-damage model, where the yield surface is defined by the Rankine criterion. The equivalent plastic strain is identical to the scalar plastic strain in 1D. Consequently, the stress-strain relation is given by

$$\sigma = e^{-\frac{\varepsilon^p}{\varepsilon_f}} E\left(\varepsilon - \varepsilon^p\right). \tag{3.55}$$

The specific fracture energy density g_f is then given by

$$g_f = \int_{\varepsilon=0}^{\infty} \sigma \, d\varepsilon = \int_{\varepsilon=0}^{f_{ct}/E} \sigma \, d\varepsilon + \int_{\varepsilon=f_{ct}/E}^{\infty} \sigma \, d\varepsilon \tag{3.56}$$

$$= \int_{\varepsilon=0}^{f_{ct}/E} E\varepsilon \, d\varepsilon + \int_{\varepsilon=f_{ct}/E}^{\infty} e^{-\frac{\varepsilon^p}{\varepsilon_f}} E\frac{f_{ct}}{E} \, d\varepsilon \tag{3.57}$$

$$= \frac{f_{ct}^2}{2E} + f_{ct}\varepsilon_f, \tag{3.58}$$

where the relation $(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^p) = \boldsymbol{\varepsilon}_0 = \frac{f_{ct}}{E}$ for the plastic region has been used. Similar to section 3.2.1, the specific fracture energy density g_f can be expressed as a function of the specific fracture energy G_f by $G_f = g_f l_{eq}$. Consequently, the parameter ε_f can be expressed as

$$\varepsilon_f = \frac{G_f}{l_{eq} f_{ct}} - \frac{f_{ct}}{2E}. \tag{3.59}$$

It is to be noted that for the investigated examples the influence of the elastic part $f_{ct}/(2E)$ is small. The equivalent length has been chosen similar to Eq. (3.11). Additionally, an axisymmetric formulation has been implemented. In this situation, the equivalent length depends on the direction of failure - in the plane direction it can be chosen similar to the plane stress assumption, whereas in the hoop direction it is given by $l_\theta = 2\pi y$, where y is the distance of the integration point to the axis of symmetry.

For mixed loading situations in an axisymmetric calculation (hoop stresses as well as stresses in the plane), an interpolation of the equivalent length has been performed, following a procedure similar to [107].

An ellipse is defined with the principal radii given as the equivalent length in the plane l_p and in the hoop direction l_θ. The interpolation angle is defined by the ratio between the plastic strain in hoop direction and the principal plastic strain in plane direction.

$$\tan\beta = \frac{\varepsilon_{zz}^p}{\varepsilon_I^p} \tag{3.60}$$

$$\varepsilon_I^p = \frac{\varepsilon_{xx}^p + \varepsilon_{yy}^p}{2} + \sqrt{\frac{\left(\varepsilon_{xx}^p - \varepsilon_{yy}^p\right)^2}{4} + \left(\varepsilon_{xy}^p\right)^2} \tag{3.61}$$

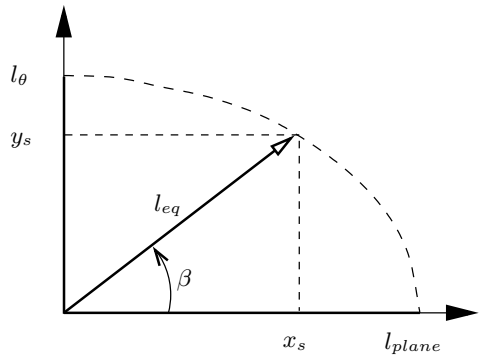


Figure 3.5 Interpolation of the equivalent length for axisymmetric calculations.

The equivalent length is calculated as the length from the origin to the ellipse in the direction of β, which is illustrated in Fig. 3.1. Solving the system of equations

$$1 = \left(\frac{x_s}{l_p}\right)^2 + \left(\frac{y_s}{l_\theta}\right)^2 \tag{3.62}$$

$$y_s = \tan\beta \; x_s \tag{3.63}$$

gives the parameter x_s and y_s and finally the equivalent length l_{eq}:

$$x_s = \sqrt{\frac{1}{\dfrac{1}{l_p^2} + \dfrac{\tan^2\beta}{l_\theta^2}}} \tag{3.64}$$

$$l_{eq} = \sqrt{x_s^2 + y_s^2}. \tag{3.65}$$

3.3.4 Nonlocal damage formulation

Similar to the isotropic damage model, the mesh sensitivity of the local formulation is reduced by introducing the factor l_{eq} that accounts for the thickness of the local zones of damage. However, the zone of damage decreases with the meshsize and, in general, damage localizes only in one row of elements. Furthermore, the calculation of the equivalent length is not straightforward, since for different orientations of the element with respect to the loading direction a modification of the equivalent length would be required. In order to circumvent these drawbacks, a nonlocal formulation is used. In nonlocal formulations, certain variables are replaced by their nonlocal average. If t is a local field, its nonlocal counterpart is given by

$$\bar{t}(\boldsymbol{x}) = \int_V w(\boldsymbol{x}, \boldsymbol{\xi}) \; t(\boldsymbol{\xi}) \; d\xi, \tag{3.66}$$

where $w(\boldsymbol{x}, \boldsymbol{\xi})$ is a given weight function. For an isotropic weight function, the weight depends only on the distance r between source and target point. Furthermore, the weight function is scaled, so that a uniform field t is identical to its nonlocal counterpart $\bar{t}$. Similar to Eq. (3.14)

for the isotropic damage model, the rescaled bell-shaped weighting function is used. [95] investigated nonlocal models with respect to their capability to describe the material until final failure. For certain nonlocal formulations, locking phenomena occurred and the material response could only be correctly approximated close to the peak of the load-displacement curve. For nonlocal models of the form

$$\boldsymbol{\sigma} = (1 - \omega(\bar{\boldsymbol{\varepsilon}}^p))\, \boldsymbol{C}\boldsymbol{\varepsilon} \tag{3.67}$$

no locking phenomena were found. In this context, a second variant was investigated where not the strain is averaged, but the equivalent strain. No substantial difference in the results has been realized and due to numerical reasons, the second variant has been finally applied, since for the nonlocal averaging only a scalar variable has to be computed, whereas in Eq. (3.67) a field of up to six variables (depending on the number of plastic strain components) has to be averaged. In the nonlocal formulation, the evolution law is replaced by

$$\omega = 1 - e^{-\frac{\hat{\kappa}}{\varepsilon_f}} \tag{3.68}$$

$$\hat{\kappa} = m\bar{\kappa} + (1 - m)\kappa \tag{3.69}$$

where κ is the local equivalent plastic strain and $\bar{\kappa}$ is calculated using Eq. (3.66) with the bell-shaped weighting function. For $m = 1$, this nonlocal formulation corresponds to the standard nonlocal averaging of the integral type. For $m > 1$ an overnonlocal formulation is obtained. In the nonlocal formulation, energy is not only dissipated in a single element, but the crack stretches over a width of approximately $2R$.

The general procedure for the calculation of the stresses at an integration point given the previous history variables and the current strain state is summarized in Algorithm 3.1.

3.3.4.1 Determination of the material parameters for the nonlocal formulation

In order to determine the influence of the parameter ε_f, a uniaxial tensile test is performed, whose geometric setup is illustrated in Fig. 3.6. A plane strain formulation is used, where the Poisson's ratio is set to zero to obtain a uniaxial test. The specimen length is 17mm, which is meshed with quadratic plane elements (9 nodes). The thickness t_e, the length l_e and the height h_e of each element are constant with $t_e = l_e = h_e = 1mm$. Only the lower and upper nodes in the plane of symmetry are shifted towards the center of the element in order to obtain a weakened section, where damage starts to localize. A shift of 0.025mm was required in order to obtain a localized solution for all investigated scenarios. In Fig. 3.7, the distribution of damage, the plastic strains and the dissipated inelastic energy density is shown.

In the upper part, the solution for $m = 1$, which is the standard nonlocal formulation, is plotted. It is observed that the plastic strains in Fig. 3.7(b) are constant within the center element and almost zero elsewhere. In a similar way, inelastic energy is primarily dissipated in the center element, which is due to the large plastic strains within the element. It is further observed that the distribution of plastic strains and dissipated energy is almost independent of the influence radius R in the nonlocal formulation. The damage distribution in Fig. 3.7(a),

Algorithm 3.1 Closest point projection for multisurface plasticity with damage

1. Compute elastic trial state
$$\begin{aligned} \bar{\boldsymbol{\sigma}}^{trial} &= \boldsymbol{C}\left(\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}^p_n\right) \\ f^{trial}_{\alpha,n+1} &= f^p_\alpha(\bar{\boldsymbol{\sigma}}^{trial}) \end{aligned}$$

2. Determine set of active yield functions $\mathcal{J}$
$$\mathcal{J} = \left\{ f^p_\alpha : f^{trial}_{\alpha,n+1} \geq -tol_1 \right\}$$
if $\mathcal{J} = \emptyset$ **then**
Set $(*)_{n+1} = (*)^{trial}_{n+1}$ and **Return**
else
$$\begin{aligned} \boldsymbol{\varepsilon}^p_{n+1} &= \boldsymbol{\varepsilon}^p_n \\ \Delta\gamma_\alpha &= 0 \quad \forall \alpha : f^p_\alpha \epsilon \mathcal{J} \end{aligned}$$

3. Evaluate flow rule and determine stress
$$\begin{aligned} \bar{\boldsymbol{\sigma}}^{(t+1)}_{n+1} &= \boldsymbol{C}\left(\boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}^{p,(t)}_{n+1}\right) \\ \boldsymbol{R}^{(t+1)} &= \boldsymbol{\varepsilon}^p_n - \boldsymbol{\varepsilon}^{p,(t)}_{n+1} + \textstyle\sum_{\alpha\epsilon\mathcal{J}} \Delta\gamma_\alpha \dfrac{\partial f^p_\alpha(\bar{\boldsymbol{\sigma}}_{n+1})}{\partial \bar{\boldsymbol{\sigma}}} \end{aligned}$$

4. Check for convergence and add influence of damage formulation
if $f^p_\alpha(\bar{\boldsymbol{\sigma}}^{(t)}_{n+1}) \leq tol_1 \forall f^p_\alpha \epsilon \mathcal{J}$ **and** $\|\boldsymbol{R}^{(t+1)}\| \leq tol_2$ **then**
if local formulation **then**
$$\hat{\kappa}_{n+1} = \kappa_{n+1} = \|\boldsymbol{\varepsilon}^{p,(k+1)}_{n+1}\|$$
else {nonlocal formulation}
$$\begin{aligned} \bar{\kappa}_{n+1} &= \textstyle\int_\Omega \alpha(\boldsymbol{x},\boldsymbol{\xi}) \|\boldsymbol{\varepsilon}^{p,(k+1)}_{n+1}(\boldsymbol{\xi})\| \, d\boldsymbol{\xi} \\ \hat{\kappa}_{n+1} &= m\bar{\kappa}_{n+1} + (1-m)\kappa_{n+1} \end{aligned}$$
Calculate damage variable and final stress
$$\begin{aligned} \omega &= 1 - \exp(-(\hat{\kappa}_{n+1})/\kappa_d) \\ \boldsymbol{\sigma}_{n+1} &= (1-\omega)\bar{\boldsymbol{\sigma}}^{(t+1)}_{n+1} \end{aligned}$$
Return

5. Calculate tangent modulus and increment to plastic multiplier
$$\begin{aligned} \boldsymbol{\Sigma} &= \left[\boldsymbol{C}^{-1} + \textstyle\sum_{\alpha=1}^m \Delta\gamma_\alpha \dfrac{\partial^2 f^p_\alpha(\bar{\boldsymbol{\sigma}}_{n+1})}{\partial \boldsymbol{\sigma}^2}\right]^{-1} \\ \Delta\Delta\boldsymbol{\gamma} &= \left\{\partial_\sigma \boldsymbol{f}(\bar{\boldsymbol{\sigma}}_{n+1})\boldsymbol{\Sigma}\left[\partial_\sigma \boldsymbol{f}(\bar{\boldsymbol{\sigma}}_{n+1})\right]^T\right\}^{-1}\left\{\boldsymbol{f}^{(t)}(\bar{\boldsymbol{\sigma}}_{n+1}) - \partial_\sigma \boldsymbol{f}(\bar{\boldsymbol{\sigma}}_{n+1})\boldsymbol{\Sigma}\boldsymbol{R}^{(t)}\right\} \end{aligned}$$

6. Compute plastic multiplier and check consistency condition
for all $\alpha : f^p_\alpha \epsilon \mathcal{J}$ **do**
if $\Delta\boldsymbol{\gamma}^{(t)} + \Delta\Delta\boldsymbol{\gamma} > 0$ **then**
$$\Delta\boldsymbol{\gamma}^{(t+1)} = \Delta\boldsymbol{\gamma}^{(t)} + \Delta\Delta\boldsymbol{\gamma}$$
else
remove f^p_α from set of active yield functions $\mathcal{J}$
if set of active yield functions $\mathcal{J}$ is modified **then**
Goto 3

7. Update plastic strain and consistency parameter
$$\boldsymbol{\varepsilon}^{p,(t+1)}_{n+1} = \boldsymbol{\varepsilon}^{p,(t)}_{n+1} + \boldsymbol{C}^{-1}\boldsymbol{\Sigma}\left[\boldsymbol{R}^{t+1}_{n+1} + \textstyle\sum_{\alpha: f^p_\alpha \epsilon \mathcal{J}} \Delta\Delta\gamma_\alpha \partial_\sigma f^p_\alpha(\bar{\boldsymbol{\sigma}}^{t+1}_{n+1})\right]$$
Goto 3

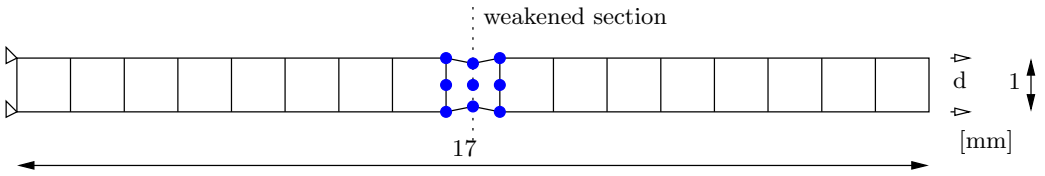


Figure 3.6 Uniaxial tensile test with a weakened section in the center element.

where the horizontal axis is scaled with $1/R$, is almost identical for all radii. By contrast, the overnonlocal formulation with $m = 2$ gives a broad zone, where energy is dissipated. The size of the zone has almost twice the size of the influence radius R. In a similar way, plastic strains do not localize in a single element layer. A phenomenon, physically not interpretable, is the damage distribution in the weakened section. Due to the overnonlocal formulation, the parameter $\hat{\kappa}$ might become negative in this section, since the nonlocal $\bar{\kappa}$ is smaller than the local κ. In this case, $\hat{\kappa}$ is set to zero, which leads to zero damage in the weakened section. Only when the neighboring sections start to dissipate energy, the nonlocal equivalent plastic strain within the weakened section increases and, consequently, damage is obtained.

The determination of the nonlocal radius is a severe problem. Many researchers claim that this nonlocal radius R is a material property. This seems to be reasonable, since the size of the fracture process zone is related to the intrinsic material length of the material. [108] tried to estimate the characteristic length by comparing tensile tests of concrete casted into steel pipes, which prevents localization, with unrestrained tests. For their damage model, they found a characteristic length of 2.7 multiplied with the aggregate diameter. [109] investigated the size of the fracture process zone from mesoscale simulations using lattice models in order to determine an appropriate nonlocal radius for different weighting functions. For concrete, it was found that the size of the fracture process zone relates to the size of the largest particles in the model. Fixing the nonlocal radius as a material property, the parameter ε_f is directly related to the specific fracture energy of the material. As a consequence, two of the three parameters ε_f, R and G_f can be defined by the user, whereas the third parameter can directly be calculated. In order to avoid a material parameter such as the physically difficult to interpret parameter ε_f, it was described as a function of the nonlocal radius R and the specific fracture energy G_f. The nonlocal radius was chosen in the range of the size of the smallest particle explicitly considered in the simulation, thus approximately in the range of the largest particle homogenized in the matrix material, which corresponds well to the results in [109].

Using a uniaxial tensile test as described in Fig. 3.6, the parameter ε_f can be expressed as a function of the nonlocal radius. For m=1, the weighting function is given by

$$\alpha(r) = \left[1 - \left(\frac{r}{R}\right)^2\right]^2. \tag{3.70}$$

In order to rescale the weighting function, the integral over the domain of influence is performed:

$$\alpha_{tot} = 2\int_{r=0}^{R} \alpha(r)dr = \frac{16}{15}R. \tag{3.71}$$

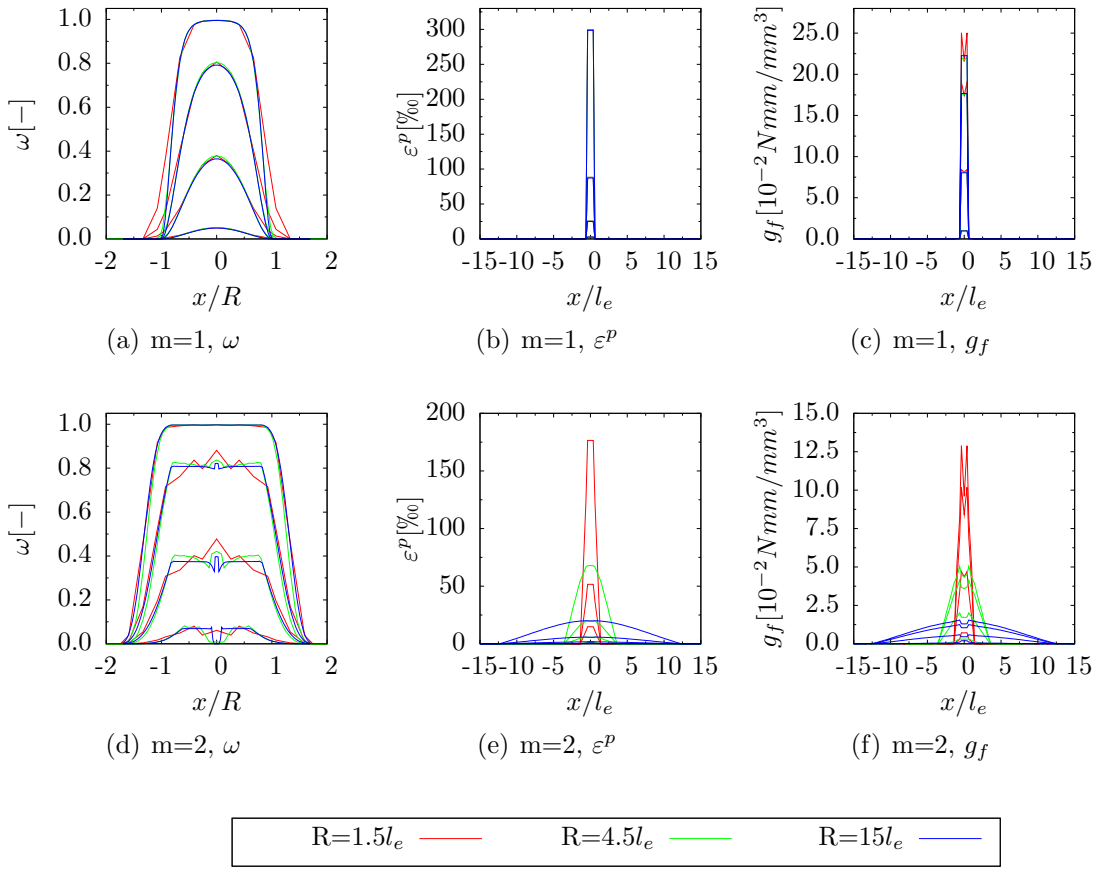


Figure 3.7 Influence of nonlocal averaging (m=1 and m=2) on the distribution of damage, inelastic energy and local plastic strain for different load levels (d=0.003mm, d=0.01mm, d=0.03mm and d=0.1mm) and a mesh size of 0.33mm.

The plastic strains localize within the center element with element length l. The averaged nonlocal plastic strain in the center element is then given by

$$\bar{\kappa} = 2 \int_{r=0}^{l/2} \kappa_m \frac{\alpha(r)}{\alpha_{tot}} dr, \tag{3.72}$$

where κ_m is the local plastic strain in the center element which is assumed to be constant. This can be simplified to

$$\bar{\kappa} = \frac{3}{256} \frac{\kappa_m l^5}{R^5} - \frac{5}{32} \frac{\kappa_m l^3}{R^3} + \frac{15}{16} \frac{\kappa_m l}{R}. \tag{3.73}$$

For a nonlocal radius larger than the element size, this reduces to

$$\bar{\kappa} = \frac{15}{16} \frac{\kappa_m l}{R}. \tag{3.74}$$

The energy dissipated within the center element is given by

$$W = \int_{u=0}^{\infty} F(u)\, du \tag{3.75}$$

$$= \int_{\varepsilon=0}^{\infty} \sigma(\varepsilon) A_f l_{av}\, d\varepsilon = \int_{\varepsilon=0}^{\varepsilon_0} \sigma(\varepsilon) A_f l_{av}\, d\varepsilon + \int_{\varepsilon=\varepsilon_0}^{\infty} \sigma(\varepsilon) A_f l_{av}\, d\varepsilon \tag{3.76}$$

$$= \int_{\varepsilon=0}^{\varepsilon_0} E\varepsilon A_f l_{av}\, d\varepsilon + \int_{\kappa_m=0}^{\infty} e^{-\frac{\bar{\kappa}}{\varepsilon_f}} E(\varepsilon - \varepsilon^p) A_f l_{av}\, d\kappa_m \tag{3.77}$$

where A_f is the fracture surface and $l_{av} = l$ (element length) for plane stress and $l_{av} = 2\pi y$ for failure in the ring direction. The fracture energy is defined as

$$G = \frac{W}{A_f} \tag{3.78}$$

$$= \frac{l_{av} f_{ct}^2}{E} + \frac{16}{15}\frac{R}{l}\varepsilon_f l_{av} E\varepsilon_0, \tag{3.79}$$

where for perfect plasticity the elastic strain is given by $\varepsilon_0 = \varepsilon - \varepsilon^p$ for $\varepsilon^p > 0$ and $E\varepsilon_0 = f_{ct}$. It follows, that

$$\varepsilon_f = \frac{15l\left(G - \dfrac{l_{av} f_{ct}^2}{E}\right)}{16 R l_{av} f_{ct}}. \tag{3.80}$$

For the investigated examples, the influence of the elastic part energy is negligible (with $G \approx 0.1\frac{Nmm}{mm^2}$, $f_{ct} \approx 4\frac{N}{mm^2}$, $E \approx 30000\frac{N}{mm^2}$ and $l < 5mm$, which finally gives:

$$\varepsilon_f = \frac{15G}{16 f_{ct} l_{eq}} \begin{cases} l_{eq} = R & \text{for plane stress and full 3D simulations} \\ l_{eq} = \dfrac{R\, 2\pi y}{l} & \text{for hoop stresses.} \end{cases} \tag{3.81}$$

Similar to the axisymmetric local formulation, the equivalent length is a function of the distribution of plastic strains within the material. The interpolation is performed similar to Eq. (3.60), where the local plastic strain is replaced by its nonlocal counterpart.

In a similar way, the parameter ε_f can be calculated for the overnonlocal formulation with $m = 2$. The distribution of the inelastic energy and the plastic strains can be approximated in a first approximation by a hat function. The peak values are given by E_m and κ_m, reducing to zero at $\pm R$. As a result, the average equivalent plastic strain at the peak point $\bar{\kappa}$ can be expressed as

$$\bar{\kappa} = 2\int_{r=0}^{R} \kappa_m \left(1 - \frac{r}{R}\right) w(r)\, dr \tag{3.82}$$

with the weighting function

$$w(r) = \frac{\left(1 - \frac{r^2}{R^2}\right)^2}{2\int_{r=0}^{R}\left(1 - \frac{r^2}{R^2}\right)^2}. \tag{3.83}$$

Substitution of Eq. (3.71) gives

$$\bar{\kappa} = \frac{2 \cdot 15\kappa_m}{16R} \int_{r=0}^{R} \left(1 - \frac{r}{R}\right) \left(1 - \frac{r^2}{R^2}\right)^2 \, dr \tag{3.84}$$

$$= \frac{11}{16}\kappa_m. \tag{3.85}$$

The nonlocal value $\hat{\kappa}$ is used in the softening relation.

$$\hat{\kappa} = m\bar{\kappa} + (1-m)\kappa \tag{3.86}$$

$$= \frac{3}{8}\kappa_m \tag{3.87}$$

The maximum energy dissipated in the weakened element at the center can be expressed as:

$$W = \int_{u=0}^{\infty} F(u) \, du \tag{3.88}$$

$$= \int_{\varepsilon=0}^{\infty} \sigma(\varepsilon) A_f l_{av} \, d\varepsilon = \int_{\varepsilon=0}^{\varepsilon_0} \sigma(\varepsilon) A_f l_{av} \, d\varepsilon + \int_{\varepsilon=\varepsilon_0}^{\infty} \sigma(\varepsilon) A_f l_{av} \, d\varepsilon = \tag{3.89}$$

$$= \frac{A_f l_{av} f_{ct}^2}{2E} + \int_{\kappa_m=0}^{\infty} e^{-\frac{\hat{\kappa}}{\varepsilon_f}} E\varepsilon_e A_f l_{av} \, d\kappa_m \tag{3.90}$$

$$= \frac{A_f l_{av} f_{ct}^2}{2E} + f_{ct} A_f l_{av} \frac{8\varepsilon_f}{3} \tag{3.91}$$

where $l_{av} = l$ (element length) for plane stress and $l_{av} = 2\pi y$ for failure in the ring direction and A_f is the fracture surface. Using the assumption of the hat distribution for the energy it follows for the total energy

$$W_{tot} = c_f \frac{W}{l} \frac{2R}{2}, \tag{3.92}$$

with $c_f = 0.8$ a correction factor to take into account the nonlinear distribution of the energy along the nonlocal interaction radius determined from numerical simulations. The fracture energy is defined as

$$G = \frac{W_{tot}}{A_f} \tag{3.93}$$

$$= c_f \frac{R}{l} \left(\frac{l_{av} f_{ct}^2}{2E} + f_{ct} l_{av} \frac{8\varepsilon_f}{3} \right) \tag{3.94}$$

where for perfect plasticity $\varepsilon_e = \varepsilon_{tot} - \varepsilon^p$ and $E\varepsilon_e = f_{ct}$ for $\varepsilon^p > 0$. Solving Eq. (3.94) for ε_f, it follows that

$$\varepsilon_f = 3 \frac{\dfrac{Gl}{c_f R} - \dfrac{l_{av} f_{ct}^2}{2E}}{8 f_{ct} l_{av}}. \tag{3.95}$$

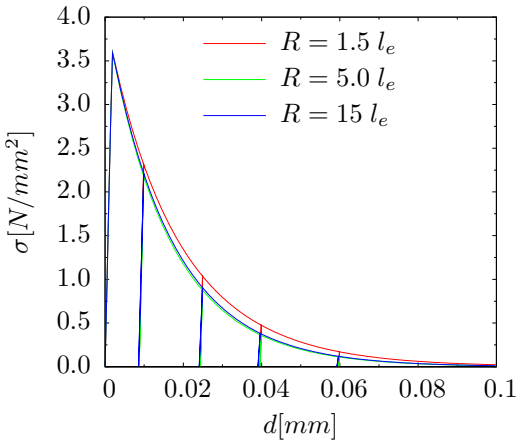


Figure 3.8 Load-displacement curve for cyclic loading test using the plasticity formulation in combination with nonlocal damage for different nonlocal influence radii R and $m = 1$.

Assuming that the term from the pre-peak elastic part can be neglected, the parameter ε_f is given by

$$\varepsilon_f = \frac{3G}{8c_f f_{ct} l_{eq}} \begin{cases} l_{eq} = R & \text{for plane stress state} \\ l_{eq} = \frac{2\pi y R}{l} & \text{for hoop stresses} \end{cases} \tag{3.96}$$

Similar to the axisymmetric local formulation, the equivalent length is a function of the distribution of plastic strains within the material. The interpolation is performed similar to Eq. (3.60), where the local plastic strain is replaced by its nonlocal counterpart.

In most of the numerical examples, the nonlocal formulation with $m = 1$ has been used. This is on the one hand due to the physically not interpretable damage distribution in the center element (lower damage in the weakened section than in the neighboring elements) and, on the other hand, due to the calculation of the material parameter ε_f, which can be, for $m = 1$, in a straightforward way related to the fracture energy, often given in experimental tests, whereas for $m = 2$ additional assumptions (hat-like distribution, correction factor) have to be used.

3.3.4.2 Load-displacement curve in tension under cyclic loading

The uniaxial test, described in Fig. 3.6, was used to investigate the behavior of the material formulation in a cyclic loading test with the standard nonlocal formulation (m=1). The corresponding load-displacement curves are given in Fig. 3.8. For comparison, different influence radii R have been used. It is to be noted that in the early region, large plastic strains occur in the center element, whereas the stiffness reduction is still moderate. Close to failure, where the transmitted stresses are negligible, a considerable stiffness reduction is obtained, accompanied by large plastic strains. This is due to the fact that the plasticity formulation is defined in the effective stress space, which means that especially close to failure, where a considerable difference between the effective and the nominal stresses is observed, large effective stresses are obtained. This further implies that the plasticity formulation has to be numerically stable also for large plastic strains. At this point, a further remark concerning the

assumption of small strains in the element formulation is required. The large plastic strains are accompanied by the isotropic damage formulation. The larger the plastic strains are, the closer the damage variable tends to one, and the element is not able to transmit any forces. Consequently, the higher the error in the small strain formulation is, the less significant it becomes.

Comparing the load-displacement curves for the different nonlocal influence radii R, it is observered that there is almost no difference. This is especially important for the relatively small influence radius $R = 1.5 l_{eq}$, since in the numerical simulation a large influence radius leads to a large bandwith and many nonzero elements in the stiffness matrix and, consequently, a large memory costs, which is tried to be avoided.

3.3.5 Nonlocal stiffness matrix

The material law is used in complex mesoscale simulations. Without the correct tangential stiffness matrix, no convergence of the numerical solution could be obtained. The algorithmic tangent modulus can be determined from

$$\boldsymbol{\sigma} = (1-\omega)\boldsymbol{C}\left(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^p\right) \tag{3.97}$$

$$\frac{d^{(i)}\boldsymbol{\sigma}}{d^{(j)}\boldsymbol{\varepsilon}} = -\frac{d^{(i)}\omega}{d^{(j)}\boldsymbol{\varepsilon}}\boldsymbol{C}\left(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^p\right) + \left(1 - {}^{(i)}\omega\right)\boldsymbol{C}\left(\frac{d^{(i)}\boldsymbol{\varepsilon}}{d^{(j)}\boldsymbol{\varepsilon}} - \frac{d^{(i)}\boldsymbol{\varepsilon}^p}{d^{(j)}\boldsymbol{\varepsilon}}\right), \tag{3.98}$$

where ${}^{(i)}$ represents the variables at integration point i. Substituting Eqs. (3.68) and (3.69) gives

$$\frac{d^{(i)}\omega}{d^{(j)}\boldsymbol{\varepsilon}} = \frac{\partial^{(i)}\omega}{\partial^{(i)}\hat{\kappa}}\left(\frac{\partial^{(i)}\hat{\kappa}}{\partial^{(i)}\bar{\kappa}}\frac{d^{(i)}\bar{\kappa}}{d^{(j)}\boldsymbol{\varepsilon}} + \frac{\partial^{(i)}\hat{\kappa}}{\partial^{(i)}\kappa}\frac{\partial^{(i)}\kappa}{d^{(j)}\boldsymbol{\varepsilon}}\right) \tag{3.99}$$

$$= \left(-\frac{1}{\varepsilon_f}\exp^{-\frac{\hat{\kappa}}{\varepsilon_f}}\right)\left(m\frac{d^{(i)}\bar{\kappa}}{d^{(j)}\boldsymbol{\varepsilon}} + (1-m)\frac{d^{(i)}\kappa}{d^{(j)}\boldsymbol{\varepsilon}}\right) \tag{3.100}$$

$$= \left(-\frac{1}{\varepsilon_f}\exp^{-\frac{\hat{\kappa}}{\varepsilon_f}}\right)\left(m\,{}^{(i)}w_j\frac{\partial^{(j)}\kappa}{\partial^{(j)}\boldsymbol{\varepsilon}^p}\frac{d^{(j)}\boldsymbol{\varepsilon}^p}{d^{(j)}\boldsymbol{\varepsilon}} + (1-m)\delta_{ij}\frac{\partial^{(j)}\kappa}{\partial^{(j)}\boldsymbol{\varepsilon}^p}\frac{d^{(j)}\boldsymbol{\varepsilon}^p}{d^{(j)}\boldsymbol{\varepsilon}}\right), \tag{3.101}$$

where ${}^{(i)}w_j$ is the weighting function of integration point (j) at integration point i, which is not symmetric any more (${}^{(i)}w_j \neq {}^{(j)}w_i$). Furthermore, δ_{ij} is the Kronecker symbol with $\delta_{ij} = 1$ for $i = j$ and 0 otherwise. All of the derivatives given in Eq. (3.101) can be computed, where $\frac{d^{(j)}\kappa}{d^{(j)}\boldsymbol{\varepsilon}^p}$ can be derived from Eq. (3.53). The elasto-plastic tangent modulus $\frac{d\boldsymbol{\sigma}}{d\boldsymbol{\varepsilon}}$ is given in Eq. (3.48). It is to be noted that the stresses in one element are a function of the strains/displacements in the neighboring elements. Consequently, the stiffness matrix of a single element is no longer quadratic, but has a rectangular shape and a nonsymmetric solver has to be used. In the numerical application, two successive calls of the material law are performed. In the first run, the local elasto-plastic tangent is calculated for all integration points and stored in a

temporary field. Within the second run, the full tangential stiffness matrix is computed. The nonlocal weights ${}^{(i)}w_j$ of the integration points are calculated in advance and stored at the integration point.

Chapter 4

Simulation of concrete on the mesoscale

4.1 Introduction

A simulation of concrete on the macroscale allows a characterization of the homogenized response on lower scales, but the true physical phenomena can only be represented in a phenomenological approach. This often requires complex material formulations with many parameters, which are often difficult to determine. The modeling of concrete on the mesoscale allows for the direct representation of the heterogenous structure [110], including e.g. the mass fraction and the grading curve of particles, the shape of the particles [111], the interfacial transition zone (ITZ) at the interface between particles and matrix material and a separate constitutive formulation for each of the individual constituents (particles, mortar matrix, ITZ).

For numerical simulations, two approaches to describe the mesostructure can be distinguished. On the one hand, real specimens can be analyzed and, by using digital image analysis, equivalent numerical models can be built [112]. On the other hand, the structure of real concrete samples on the mesoscale can be investigated using X-ray tomography and procedures for the numerical simulation are derived [113, 114]. A functional description of the grading curve and the particle shape allows the generation of random samples of the heterogenous microstructure of concrete [115].

The first mesoscale models of concrete are based on the lattice approach [116, 117, 118, 119]. According to the position in the model, a material (particle, matrix, ITZ) is assigned to any lattice. The properties of the beam elements, which might include a stochastic component, are derived from macroscopic properties such as Young's modulus or Poisson's ratio. Different approaches to model the failure of a single lattice have been investigated. One possibility is the so called tension cut-off, where the lattice is removed from the model, if the prescribed tensile strength is exceeded. However, this leads to spurious steps in the load-displacement curve each time a lattice is removed. In another approach, a linear model for the prepeak region combined with a softening function is used [120]. The primary problem of these approaches is the determination of the material parameters, since no direct relation between the properties of the lattice and the obtained macroscopic properties can be derived.

A second group are particle methods, which can be used for the simulation of cohesive granular materials [121]. Particles are simulated as discontinuous spheres, where the interaction between neighboring particles is solely determined by their relative spatial position. A similar difficult problem is here the determination of material parameters for the description of the cohesion between the particles.

A third group are continuum models [122, 123], where the mesostructure is explicitly represented. In this context, it is assumed that simple material models can be used, since the complexity of the response is additionally included in the complex heterogenous geometric description of the material. Furthermore, the nonlinear behavior of the ITZ at the interface between mortar matrix and particle can directly be simulated, e.g. with a cohesive model. Generally, a discretization with an aligned mesh is used for this purpose. In [15, 124], a procedure using the XFEM approach in combination with level sets is used to model the ITZ, which allows the application of a regular quadrilateral mesh. In [2] it is assumed that the nonlinear behavior can solely be described by a nonlinear model for the interface. For that purpose, a mesoscale model with linear elastic particles is meshed with triangular elements, and interface elements are placed between each finite element. The disadvantage of this approach is that the crack path is limited to the element edges and a strong mesh dependency is obtained.

Advantageous of all mesoscale models is the fact that the complex macroscopic material behavior can be described by simple material formulations of each constituent on the mesoscale. An example is the nonlinear prepeak region in a tensile test, which can be described on the mesoscale without any hardening variables. Furthermore, the artificial insertion of a weak point to initiate localization as e.g. in a uniaxial tensile test can be avoided, since, due to the heterogenous structure on the mesoscale, stress concentrations and, consequently, zones of local damage naturally occur.

4.2 Experimental results

Concrete is a heterogenous material whose material behavior is strongly related to its microstructure. The nonlinear behavior of concrete is caused by the development of micro cracks, which finally coalesce into a macroscopic crack. An important influence on the development of microcracks has the underlying microstructure. For normal strength concrete where the strength of the aggregates is much higher compared to the mortar matrix, failure generally is initiated at the interface between mortar matrix and aggregates, which has substantially lower material properties than the bulk material [125], e.g. a higher porosity. Consequently, three principal components can be used to characterize the mesoscale model - aggregates, mortar matrix and the interfacial transition zone.

4.2.1 Aggregates

In normal strength concrete, the mass fraction of aggregates is approximately 60-70%. The size distribution of aggregates is characterized by the grading curve, which describes the mass of aggregates that passes through sieves of different sizes. Different materials can be used as aggregates. In Table 4.1, experimental properties of aggregates from different experimental tests are illustrated.

Obviously, there is a large scatter in the data, which is, on the one hand, due to the scatter in the material itself, i.e. stones from different locations have different chemical composition and

Table 4.1 Experimental results for material parameters of aggregates : compressive strength f_c, tensile strength f_{ct}, Young's modulus E and Possion ration ν.

author	property	basalt	limestone	granite	sandstone
[126]	f_c in MPa	158	52		
	f_t in MPa	10.2	8.4		
	E in GPa	54	67		
[127]	f_c in MPa		57.5	140.1	
	ν		0.18	0.16	
	E in GPa		34.5	55.3	
[128]	f_c in MPa		147.3 (±32.77%)	179.1 (±43.45%)	102.8 (±3.39%)
	E in GPa		65.6 (±2.84%)	67.8 (±3.37%)	26.7 (±5.67%)
	ν		0.27 (±10.48%)	0.24 (±11.79%)	0.18 (±31.43%)

different material properties, and, on the other hand, due to the stochastic influence in the experiments, i.e. repeating the same experiment with the same setup will still give different results. Another influencing factor is the setup of the experimental tests, e.g. the size of the specimen has a considerable influence on the measured strength values.

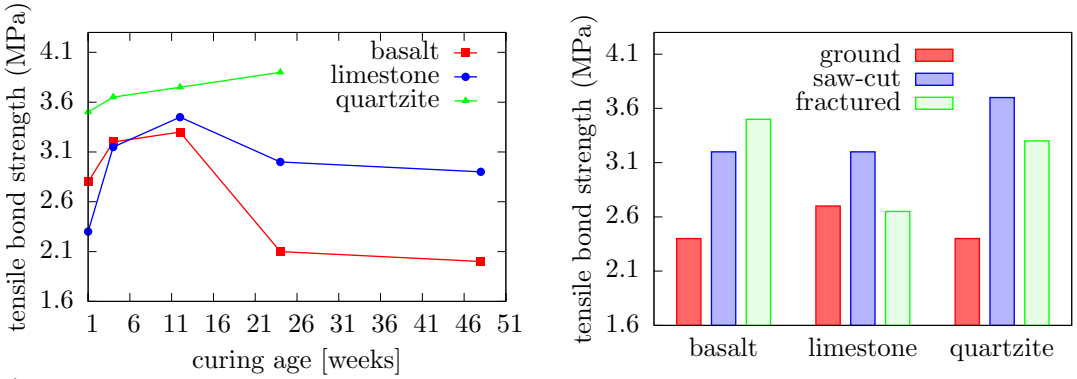
For a numerical simulation of concrete, where the exact parameters of the aggregates are not given, a rough estimate of the material properties of aggregates can be made according to Table 4.1. The influence of the strength values is, for normal strength concrete, relatively small, since the aggregates can be assumed to be linear elastic.

Another classification of aggregates is made, according to its manufacturing process, in fractured (split, crushed stone fines) and unfractured (gravel, sand), but in the numerical simulation only round, unfractured particles are simulated.

4.2.2 Mortar matrix

In this context, the mortar matrix comprises all components with a diameter less than 2mm. Obviously, the mortar matrix is a heterogenous material, which consists of pores, small aggregates and hardened cement paste, which itself is also heterogenous. For high compaction, the volume ratio of pores is about 1.5%, but for special concrete mixtures with a high freeze-thaw resistance up to 6 % of pores are possible, but from a mesoscale point of view, the mortar matrix is assumed to be homogeneous.

The properties of the matrix are essentially determined by the cement used for the concrete, the compaction and, consequently, the amount of pores, the water-cement ratio and the environmental conditions (e.g. temperature, humidity) during the solidification process. It is to be noted that the properties of the matrix evolve during the hardening procedure. In order to obtain comparable results, properties of concrete/cement paste are, in general, experimentally investigated under standardized conditions after a fixed period of time (e.g.



(a) Temporal development of bond strength for a sawed specimen [133]

(b) Effects of aggregate surface roughness [126]

Figure 4.1 Bond strength between matrix and aggregate in mode I

28 days). However, the real properties in situ might be fundamentally different. For concrete, [129, 130] observed e.g. that within 30 to 50 years the compressive strength in general doubles and, in rare cases, an increase of up to 400% compared to the value after 28 days occurred.

4.2.3 Interfacial transition zone

The interfacial transition zone (ITZ) has an essential influence on the macroscopic properties of concrete. This region with a thickness of $20-100\ \mu$m is initially characterized by a high water-cement ratio due to the reduced amount of cement grains, which results in a higher porosity. The ITZ is rather inhomogeneous and a description of the average ITZ is difficult. Examining back scattered electron images, [125] observed an air void content of approximately 30%(Vol.) at a distance of $1.2\ \mu$m from the aggregates, which reduces to 10%(Vol.) at a distance of $50\ \mu$m. [131] investigated the temporal distribution of pores using scanning electron microscopy, and found in the final state (56 days) only in the vicinity of the aggregates ($0-15\ \mu m$) a considerable increase of the porosity, whereas in the range $15-100\ \mu m$ from the aggregates the initially increased porosity is reduced due to the hydration products. On the contrary, [132] examined concrete with aggregates made from quartzite and dolomite after 3 and 100 days and pointed out that the porosity of up to 30% close to the aggregates reported by other authors is highly overestimated and no significant increase of the porosity in the vicinity of the aggregates is observed.

Experiments which quantitatively measure the mechanical properties, e.g. the bond strength, are important to develop a mechanical model of the ITZ. The results of [126, 133] are illustrated in Fig. 4.1. Accordingly, the tensile bond strength increases during the first 12 weeks. For basalt, it drops afterwards which is explained by the decomposition of feldspar due to their interaction with the hydrating cement, which produces clay minerals that swell on absorbing water. A similar phenomenon was observed for limestone, which is, according to the authors, due to the chemical reaction of limestone and cement paste. This reaction produces carbon dioxide which results in a higher porosity of the interface.

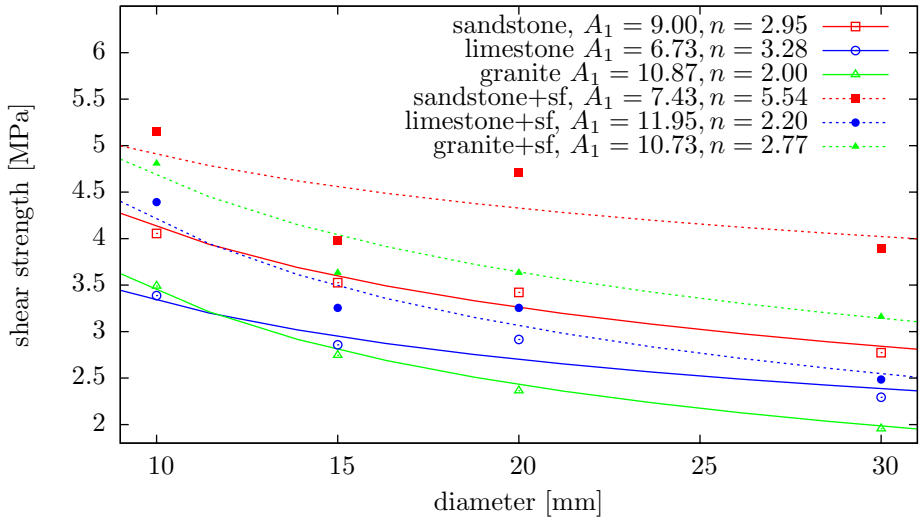


Figure 4.2 Shear bond strength (regression curve $\tau = Ad^{-1/n}$) between aggregate and matrix with and without additional silica fume [135].

[134] determined the strength of the interface between aggregates made of rough granite and different types of mortar. The measured tensile bond strength was between 0.12 and 0.56 MPa, whereas the shear strength of the interface was between 1.47 and 2.79 MPa.

In [135], the shear bond strength was determined using pullout tests of cylindrical aggregates with a diameter between 10 and 30mm. The experimental results can be approximated by $\tau = Ad^{-1/n}$, which is illustrated in Fig. 4.2. The shear strength τ in MPa is a function of the diameter d of the inclusion in [mm] and material constants A and n. As can be deduced from the results, the size effect is clearly recognizable - i.e. the smaller the specimen diameter, the higher the strength - although the number of specimens tested is too small to allow any statistical interpretation. Similar experiments have been performed by [136] for pullout tests with cylindrical specimens with a diameter of 12mm and a height of 13mm. A shear strength between 1.6N/mm^2 for coarse aggregates, 2.26N/mm^2 without aggregates and 4.71N/mm^2 with the addition of silica fume were measured. Additionally, the specific fracture energy was determined (0.007Nmm/mm^2 using coarse aggregates, 0.014Nmm/mm^2 without aggregates and 0.0659Nmm/mm^2 with silica fume), The (unrealistic) small values might be due to the fact that the specific fracture energy was not determined using the full load-displacement curve, but approximated from the peak load and the corresponding slip.

The influence of the ITZ on the macroscopic properties of concrete has been determined by [137]. Aggregates (crushed and rounded, diameter 5-7mm) were coated with bitumen emulsion, paraffin and an epoxy raisin in order to modify the surface of the aggregates and, consequently, the structure of the ITZ. Afterwards, tensile and compressive strength, Young's modulus and specific fracture energy of the concrete were determined. Some of the results are illustrated in Table 4.2. Obviously, the shape of the aggregates has a fundamental influence on the macroscopic fracture energy. For round inclusions, the fracture energy seems to be in the same range compared to the pure matrix, whereas crushed aggregates strongly increase the fracture energy. This is probably due to the fact that the width of the facture zone is larger

Table 4.2 Influence of the aggregate surface and the aggregate shape on the macroscopic properties of concrete with mean and standard deviation [137].

aggregate	coating	f_c in MPa	E in MPa	f_t in MPa	G_F in J/m^2
only matrix		71.1±0.2	34.3±0.1	4.00±0.10	67.2±2.0
crushed	bitumen emulsion	21.0±0.3	19.7±0.7	2.19±0.03	125.8±6.3
crushed	paraffin	36.1±1.3	25.6±0.6	2.99±0.10	141.1±9.0
crushed	none	73.2±2.0	33.1±0.8	4.15±0.05	136.0±4.8
round	bitumen emulsion	21.9±0.5	22.5±0.5	2.46±0.07	67.8±2.5
round	paraffin	41.5±0.8	33.7±0.6	3.11±0.04	77.0±6.7
round	none	63.8±2.6	39.8±1.0	3.93±0.05	94.7±5.7

for fractured aggregates. The tensile strength and the compressive strength of concrete with round aggregates is reduced compared to the plain matrix, whereas for crushed aggregates an increase is observed.

The most important result is the strong influence of the coating, i.e. the properties of the ITZ, on the macroscopic behavior. For a weakened interface between matrix and aggregate, all macroscopic material parameters are substantially reduced - especially for round aggregates. Under the assumption that the surface treatment mainly affects the tensile strength in the interface layer it becomes evident that the failure of the interface in tension has a fundamental influence in the macroscopic compressive strength.

In summary, it can be concluded that the material parameters for the interface and the mortar matrix show strong variations. This is due to the fact that many parameters (specimen geometry, load application, cement type, aggregate type, water-cement ratio, time between casting and testing etc.) influence the results. The influence of a single parameter, while trying to keep all other influencing factors constant, can be qualitatively estimated. Predictions of unknown material parameters for a new concrete composition with interacting influencing factors from existing experimental data are rather imprecise. Furthermore, the sample size in most of the experiments is rather small, which adds additional inaccuracies. However, the representation of the ITZ in a mechanical model seems to be essential, since cracks are initiated in and often propagate along the ITZ.

4.3 Modeling of the geometry

An important aspect of the mesoscale simulation of concrete is the exact representation of the aggregate shapes, their size distribution according to a prescribed grading curve and the spatial position and orientation of the aggregates within the specimen. Two main concepts to build a numerical mesoscale model of concrete can be distinguished. The first possibility is based on image processing techniques. Based on a X-ray computer tomography [138] or by sequential sectioning and 2-D image processing [139], a 3-dimensional voxel representation of the microstructure is obtained, which can be used in a voxel based finite element representation [140, 112, 141]. A second approach, followed in this thesis, is the artificial generation of the microstructure. [142] used polygonal and spherical inclusions to simulate concrete on the

mesoscale. For 3-dimensional models, either spheres [143, 144] or ellipsoids [145, 119, 146] are common shapes to describe the aggregates. [113] used voxel representations of real concretes to describe the size and shape of aggregates in concrete by spherical harmonics, which allows the simulation of random shapes.

In order to obtain realistic models, certain statistical characteristics of the real model, e.g. the size distribution of aggregates has to be represented in the numerical model. The size distribution of aggregates is commonly characterized by a grading curve. Grading curves for standardized concrete are defined in the design codes. A grading curve is defined by aperture sizes of a set of sieves and the corresponding mass fraction that passes through these sieves. An alternative approach is the description of the size distribution using an explicit function such as e.g. Fuller's curve. Using many different sieves, the second approach (Fuller) can almost exactly be reproduced by the first approach.

In the current implementation, an algorithm based on [147] is used. It can be divided into the generation of the correct set of aggregates and, afterwards, the placing of these aggregates into the specimen, which is commonly referred to as the set and place method.

4.3.1 Sampling of aggregates according to a size distribution

Aggregates are simplified by ellipsoids and the generation of the geometry is fully performed in 3-D. A 2-dimensional model can be obtained by cutting a slice out of the 3-dimensional model. In the local coordinate system aligned with the principal axis, the surface of an ellipsoid is defined by

$$\sum_{i=1}^{3} \left(\frac{x_i}{r_i} \right)^2 = 1, \tag{4.1}$$

where r_i are the principal radii. Without loss of generality, it is assumed that $r_1 \geq r_2 \geq r_3$. It is further assumed that an aggregate passes through a sieve, if its medium diameter $2r_2$ is smaller than the aperture of the sieve.

The radii r_1 and r_3 are determined from r_2 according to

$$r_1 = \left(1 + u_1 \frac{m-1}{m+1} \right) r_2 \tag{4.2}$$

$$r_3 = \left(1 - u_3 \frac{m-1}{m+1} \right) r_2, \tag{4.3}$$

where u_1 and u_2 are realizations of random variables U_1, U_3 which are uniformly distributed in the interval [0,1] and m is a constant that characterizes the flatness of the ellipsoids. For m=1, all radii are identical and the ellipsoids reduce to spheres, whereas for higher values the flatness increases. The assumptions of a linear distribution of r_1 and r_3 in the intervals $\left[r_2, r_2 \frac{2m}{m+1} \right]$ and $\left[r_2 \frac{2}{m+1}, r_2 \right]$ has been made for simplicity, but any other distribution can be used. The expected volume $E(V)$ of an aggregate is then given by

$$E(V) = \int_{u_1=-\infty}^{\infty} \int_{u_3=-\infty}^{\infty} \frac{4}{3} \pi^3 r_1(u_1, r_2) r_2 r_3(u_3, r_2) \, du_3 du_1. \tag{4.4}$$

Since the variables U_1 and U_3 are independent and uniformly distributed in the interval [0,1], Eq. (4.4) can be simplified to

$$E(V) = \int_{u_1=0}^{1} \int_{u_3=0}^{1} \frac{4}{3}\pi^3 r_2^3 \left(1 + u_1 \frac{m-1}{m+1}\right) \left(1 - u_3 \frac{m-1}{m+1}\right) du_3 du_1 \tag{4.5}$$

$$= \frac{4}{3}\pi^3 r_2^3 \left[1 - \left(\frac{m-1}{2(m+1)}\right)^2\right]. \tag{4.6}$$

In the grading curve, different mineral size classes i can be distinguished, each having a minimum diameter d^i_{min} and a maximum diameter d^i_{max}. From the total mass of aggregates m_{tot} and the grading curve, the mass of each mineral size class can be determined.

$$m_i = m_{tot} \left[F_m(d^i_{max}) - F_m(d^i_{min})\right], \tag{4.7}$$

where $F_m(d)$ is the ratio of mass passing through a sieve of aperture size d and the total mass. Consequently, each class can be simulated separately and the index i is omitted. The function $F_m(d)$ can be interpreted as the cumulated probability density function of the mass distribution as a function of the diameter $d_2 = 2r_2$. Assuming a linear function in the logarithmic scale between $F_m(d_{min}) = 0$ and $F_m(d_{max}) = 1$, it follows:

$$F_m(d_2) = \frac{\ln(d_2) - \ln(d_{min})}{\ln(d_{max}) - \ln(d_{min})} \quad \text{with } d_{min} \leq d_2 \leq d_{max}. \tag{4.8}$$

This can be interpreted as the accumulated probability density function of the mass as a function of the diameter d_2. The corresponding probability density function is obtained by differentiation of Eq. (4.8):

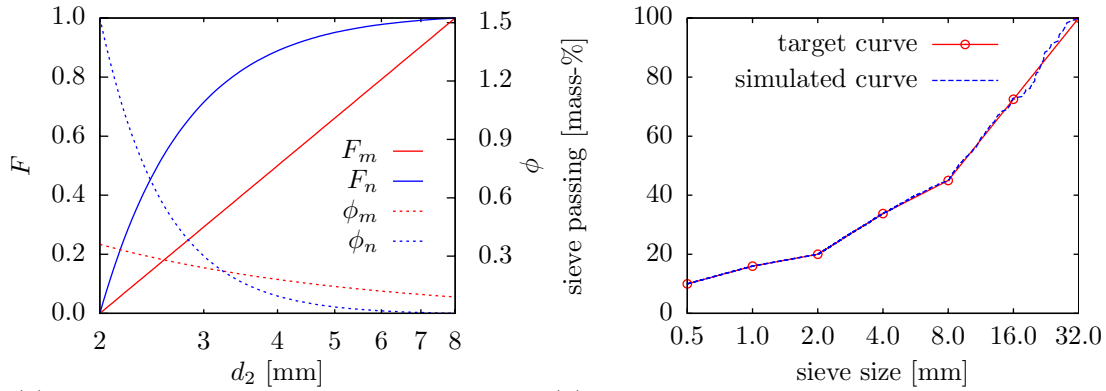
$$\phi_m(d_2) = \frac{\partial F_m(d_2)}{\partial d_2} = \frac{1}{d_2 \left[\ln(d_{max}) - \ln(d_{min})\right]}. \tag{4.9}$$

Equation (4.9) describes the mass distribution within the mineral size class. From the mass distribution, the distribution for the number of particles is derived. The mass of particles M of size d_2 is given by

$$M(d_2) = n(d_2)\rho E(V(d_2)), \tag{4.10}$$

where $n(d_2)$ is the number of particles with diameter d_2, ρ is the density of the aggregates assumed to be identical for all mineral size classes and $E(V(d_2))$ is the expected value of the volume of an aggregate with diameter d_2 given in Eq. (4.6) with $2r_2 = d_2$. Consequently, the distribution of the number of particles as a function of the radius d_2 is given by

$$\phi_n(d_2) = \frac{\dfrac{\phi_m(d_2)}{\rho E(V(d_2))}}{\displaystyle\int_{x=d_{min}}^{d_{max}} \frac{\phi_m(x)}{\rho E(V(x))}\, dx}, \tag{4.11}$$



(a) Cdf F and pdf ϕ of the distributions for mass m and number of samples n in the mineral size class 2-8.

(b) Comparison of target and simulated grading curves [123].

Figure 4.3 Sampling of particles according to a prescribed grading curve.

where the denominator corresponds to the total number of aggregates in the interval $[d_{min}, d_{max}]$. Substitution of Eq. (4.6) and simplification finally gives:

$$\phi_n(d_2) = \frac{3 d_{max}^3 d_{min}^3}{d_2^4 \left(d_{max}^3 - d_{min}^3\right)}, \tag{4.12}$$

which describes the density function for the number of ellipsoids. Integration of the density function gives the cumulated density function of the ellipsoids

$$F_n(d_2) = \int_{x=d_{min}}^{d_2} \phi_n(x) = \frac{d_{max}^3 d_{min}^3}{d_{max}^3 - d_{min}^3} \left(\frac{1}{d_{min}^3} - \frac{1}{d_2^3} \right). \tag{4.13}$$

In order to simulate particles according to the prescribed cumulative distribution function, the inversion method in section 5.2.2.1 is used, which means sampling a uniform random variable U_2 in the range [0,1] and then using the inverse function of the cumulated distribution function to obtain the corresponding diameter d_2:

$$d_2 = \frac{d_{max} d_{min}}{\sqrt[3]{u_2 d_{min}^3 + (1 - u_2) d_{max}^3}}. \tag{4.14}$$

The procedure is illustrated for the mineral size class [2,8] in Fig. 4.3(a). Note that F_m is a linear function in the logarithmic scale, whereas F_n is nonlinear. In a similar way, it is observed that the density function ϕ_m is much smaller than ϕ_n for small diameters d_2. This is due to the fact that, for a smaller diameter d_2, in order to have the same mass proportion a much higher number of samples is required.

The take process is summarized in Algorithm 4.1. Starting with the largest mineral size class, particles are generated by sampling random numbers u_1, u_2 and u_3 in the interval [0,1]. From these samples, the principal diameters/radii of the particle using Eqs. (4.14), (4.2) and (4.3) are calculated. The procedure is repeated until the mass m_i is exceeded. The difference between the simulated mass $\bar{m}_i$ and the target mass m_i is subtracted from the next mineral size class in order to ensure that the total simulated mass is almost identical to its target value.

Algorithm 4.1 Sampling of particles according to a prescribed grading curve

for each mineral size class i starting with the largest particles **do**

calculate mass fraction; for $i > 1$, substract difference from previous class

$$m_i = m_{tot}\left[F_m(d^i_{max}) - F_m(d^i_{min})\right] - (\bar{m}_{i-1} - m_{i-1})$$

repeat

Sample random numbers u_1, u_2 and u_3 in the interval [0,1]

Calculate principal diameters of particle

$$d_2 = \frac{d_{max}d_{min}}{\sqrt[3]{u_2 d^3_{min} + (1-u_2)d^3_{max}}}$$
$$d_1 = \left(1 + u_1\frac{m-1}{m+1}\right)d_2$$
$$d_3 = \left(1 - u_3\frac{m-1}{m+1}\right)d_2$$

Calculate volume and mass of particle

$$\bar{m}_i = \bar{m}_i + \rho\frac{1}{6}\pi^3 d_1 d_2 d_3$$

until $\bar{m}_i > m_i$

4.3.2 Placing the aggregates

After having created the particles for a prescribed volume, the particles are placed into the specimen at a random position and with random orientation, which are assumed to be uniformly distributed in their domain. For a 3-dimensional simulation, this corresponds to three coordinates and three orientation angles. In order to avoid overlapping between aggregates and the boundary of the domain and overlapping between neighboring particles, separation checks have to be performed. From the computational point of view, the most expensive procedure is the separation between particles, which is due to the large number of particles considered. At first, the bounding boxes of the ellipsoids are checked and in the case of their intersection, an efficient separation check for ellipsoids developed by [148] is applied. The representation of two ellipsoids $\mathcal{A}$ and $\mathcal{B}$ from Eq. (4.1) is rewritten as

$$\mathcal{A} : 0 = \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} = \frac{x_1^2}{(r_1^{\mathcal{A}})^2} + \frac{x_2^2}{(r_2^{\mathcal{A}})^2} + \frac{x_3^2}{(r_3^{\mathcal{A}})^2} - 1 \tag{4.15}$$

$$\mathcal{B} : 0 = \boldsymbol{x}^T\boldsymbol{B}\boldsymbol{x} = (x_1 - x_1^c)^2 + (x_2 - x_2^c)^2 + (x_3 - x_3^c)^2 - r^2 \tag{4.16}$$

with $\boldsymbol{x} = (x_1, x_2, x_3, 1)$ being homogeneous coordinates,

$$A = \text{diag}(1/(r_1^{\mathcal{A}})^2, 1/(r_2^{\mathcal{A}})^2, 1/(r_3^{\mathcal{A}})^2, -1) \tag{4.17}$$

$$B = \begin{bmatrix} 1 & & & -x_1^c \\ & 1 & & -x_2^c \\ & & 1 & -x_3^c \\ -x_1^c & -x_2^c & -x_3^c & -\left(r^{\mathcal{B}}\right)^2 + x_1^c + x_2^c + x_3^c \end{bmatrix} \tag{4.18}$$

in a coordinate system centered at the origin of $\mathcal{A}$ and scaled in such a way that $\mathcal{B}$ is represented by a sphere centered at (x_1^c, x_2^c, x_3^c) with radius $r^{\mathcal{B}}$. The characteristic polynomial is defined as

$$f(\lambda) = \det(\lambda \boldsymbol{A} + \boldsymbol{B}), \tag{4.19}$$

and $f(\lambda)$ is called the characteristic equation. [148] showed that two ellipsoids $\mathcal{A}$ and $\mathcal{B}$ can be separated by a plane if and only if the characteristic equation has two distinct positive roots. It is not required to exactly determine the roots, but the existence (e.g. using Sturm-sequences [149]) of two positive roots is sufficient to conclude that the ellipsoids are separated.

An important aspect is the enlargement of all radii of an ellipsoid by $\Delta r = \epsilon r_2$, where ϵ being a user-defined constant. This approach is physically motivated by [150], who showed that aggregates are surrounded by a thin film of mortar. From the numerical point of view, this procedure is essential to define a minimum distance between ellipsoids and, consequently, obtain non-distorted elements in the meshing procedure.

A further considerable improvement of the speed of the separation checks can be obtained by placing the particles in boxes with a constant edge length [151]. If an ellipsoid is to be placed into a certain box, only the ellipsoids already being placed in this box have to be considered. Especially for a large number of particles, this approach significantly reduces the number of separation checks and, consequently, the calculation time.

The randomness in the spatial location and the orientation of the particles reflects the stochastic character of the arrangement of particles in a real model. Different realizations of the particle arrangement can be obtained by modifying the seed of the random number generator used for the generation of the random locations and orientations, thus the variability of the response with respect to varying particle arrangement can be investigated.

In order to generate a 2-dimensional model, a slice of the 3-dimensional model is used, where ellipses with a radius smaller than a prescribed threshold related to the mesh size are removed.

4.4 Meshing

Once the geometrical description of the particle distribution is calculated, a meshing of the geometry is required in order to obtain a numerical FE model. Different approaches have been investigated.

In the first approach, the boundary of the ellipses/ellipsoids coincides with the element edges (aligned mesh) and an unstructured mesh with triangular/tetrahedral elements is created using the software packages Triangle [152] or Gmsh [153]. This approach has the disadvantage that for a large number of particles, the mesh generation procedure is time consuming and/or does not work at all.

For this reason, a simplified version has been implemented that combines regular and unstructured mesh generation. In a first step, a regular mesh with quadrilateral elements is created which does not consider the mesostructure of the model. For many geometries, this regular mesh can be created, even though in the current examples only rectangular domains are considered. In a second step, all elements which have a neighboring element that is intersected by an ellipse/ellipsoid are deleted. For each particle, a region composed of elements to be deleted is obtained, whereas the regions for different particles may overlap. By unification of all regions sharing an element, a disjoint set of regions is obtained. Each of these regions can be meshed similarly to the first approach with triangular/tetrahedral elements. The advantage of this approach is the small number of different domains or holes to be considered. Actually, the meshing problem with many inclusions is decomposed into several meshing problems each having a small number of inclusions.

A third approach, already presented in the first chapter of the thesis, is the XFEM, where the element edges do not coincide with the material edges. Apart from the calculation of the integration zones, no mesh generation procedure is required. From the mesh generation point of view, this approach is faster compared to the previously discussed methods. However, due to the introduction of enrichment functions with additional degrees of freedom (DOF), the total number of DOF's and, consequently, the computational effort is increased. An example for the different discretization schemes is given in section 4.5.1.

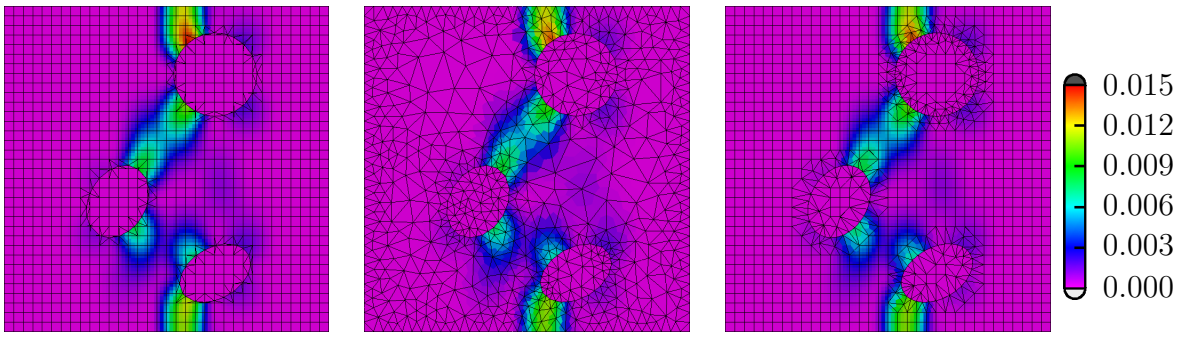
4.5 Numerical results

4.5.1 Aggregates in matrix with different discretization schemes

In a first simple example, a tensile test with only three particles under plane strain conditions is performed in order to compare the different discretization techniques. Particles are assumed to be linear elastic with Young's modulus and Poisson's ratio given by $E = 54 \cdot 10^3 \frac{\mathrm{N}}{\mathrm{mm}^2}$ and $\nu = 0.2$. The matrix is simulated using the combined damage-plasticity model presented in detail in section 3.3 with Young's modulus, Poisson's ration, fracture energy, tensile strength, uniaxial and biaxial compressive strength given by $E = 20 \cdot 10^3 \frac{\mathrm{N}}{\mathrm{mm}^2}$, $\nu = 0.18$, $f_{ct} = 3.2 \frac{\mathrm{N}}{\mathrm{mm}^2}$, $f_{ck} = 43 \frac{\mathrm{N}}{\mathrm{mm}^2}$ and $f_{ck,2} = 44.9 \frac{\mathrm{N}}{\mathrm{mm}^2}$. The standard nonlocal formulation with $\mathrm{m} = 1$ and a nonlocal interaction radius of $R = 1\mathrm{mm}$ has been used. The edge length of the model concrete specimen is 10mm with an element length of approximately 0.3mm. The left boundary is fixed in horizontal direction. A single node is fixed in vertical direction to prevent rigid body motions. At the right boundary, displacements d are applied in horizontal direction.

It is to be noted that in almost all examples quadratic elements have been used. This is due to the fact that linear triangular/tetrahedral elements can only approximate a constant strain state in a single element. In order to obtain the same accuracy in the solution, a much finer mesh and, consequently, much more DOF's compared to the quadratic approach were required.

First of all, it is observed that the computing time for the discretization with the XFEM approach is much longer compared to the aligned discretization. This is due to the fact that

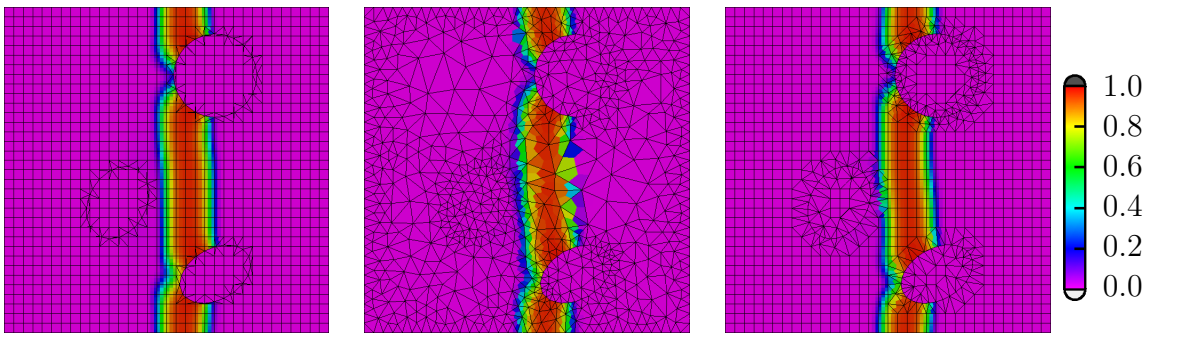


(a) XFEM (b) FEM free mesh (c) FEM regular mesh

Figure 4.4 Damage distribution at the peak load

the elements cut by the interface are decomposed into triangular integration cells, which results in much more integration points per element. The reason is due to the nonlocal formulation, where the stiffness at a given point depends on all other integration points within the nonlocal radius (assuming the point is in the inelastic regime). A second influencing factor is the number of DOF's. In the unstructured FE mesh, only 6167 DOF's were required, whereas in the regular FE mesh 10210 DOF's and in the XFEM approach 10258 DOF's have been used. Obviously, only a small number of additional enriched degrees freedom is added (compared to the regular FE mesh). The advantage of the unstructured mesh is the ability to model a varying node density, i.e. a high density close to the aggregates and a lower value in between. However it is to be noted that the variation of the element size within the same model should only be moderate in order to avoid having only a single element within the nonlocal interaction radius or too many elements, which is only a numerical problem due to the increased bandwidth of the stiffness matrix.

The obtained load-displacement curves for all three discretization techniques are almost identical, for which reason a plot of the curve has been omitted. The damage distribution at the peak load with $d = 0.0018$mm is illustrated in Fig. 4.4 and at the final displacement d = 0.1mm in Fig. 4.5. It is observed that all methods yield approximately the same damage distribution. Differences can be contributed to the different number and position of integra-



(a) XFEM (b) FEM free mesh (c) FEM regular mesh

Figure 4.5 Damage distribution at ultimate failure

tion points. It is to be noted that no interpolation of the integration point data (damage) is performed - each integration cell is filled with the color corresponding to the damage value of its integration point. For certain setups it might happen that modifications of the discretization (different discretization method or slightly varying mesh size) lead to completely different crack patterns. This is due to the fact that, in a mesoscale simulation, the crack path is strongly related to the particle distribution, and often bifurcation points occur with different crack paths being energetically almost equivalent. However, this is only a minor problem, since for large specimens, where the maximum size of the particles is small compared to the specimen dimensions, no significant difference is expected. For small specimens, only a stochastic information can be obtained, where the same specimen has to be computed several times with different particle distributions. Similarly, the influence of the meshing is removed by the averaging procedure.

An aspect strongly related to the existence of many bifurcation points are the problems in the numerical solution procedure. At a bifurcation point, at least one eigenvalue is zero. Numerically, this eigenvalue will, in general, be small, but nonzero which still allows the application of the Newton-Raphson scheme. However, the convergence speed will decrease. At these points, the application of the line search procedure is fundamental to obtain a solution, but sometimes no convergence can be obtained at all.

4.5.2 Compression test 3D

In a second example, a compression test of a concrete cube is performed in 3D and compared to experimental data from [154]. The experimentally tested concrete cube has a side length of 100mm. In the numerical simulation, a concrete cube with a side length of 25mm has been used and the results were scaled for comparison. The full cube could not be simulated numerically due to the large number of DOF's and the mesh creation in 3D. The upper and lower boundary conditions correspond to high friction between the cube and the loading platens. According to a prescribed grading curve, 39 ellipsoids in the class 4-8mm were placed into the specimen, and 230 ellipsoids in the class 2-4mm, which is illustrated in Fig. 4.6. The matrix is assumed to include all the particles smaller than 2mm. The material parameters used in the simulation are summarized in Table 4.3. It is important to mention that no parameter fitting has been performed. The fracture energy of the matrix was set to the experimental fracture energy of the macroscopic concrete specimen, which is only an approximation. Due to the increased fracture surface in a mesoscale tensile test, the fracture energy of the matrix is slightly smaller compared to the macroscopic model. Due to lack of additional information, the fracture energy and the tensile strength of the interface were chosen to be a multiple of the corresponding parameters of the matrix. The factor was chosen to be 90%, which assures that the cracks around the particles propagate along the interface elements. If cracks around the particle develop in the matrix with the crack being almost parallel to the interface between matrix and particle, boundary effects due to the renormalization of the nonlocal weighting function occur, and the assumption of energy dissipation in the full range of the nonlocal radius is not valid any more. The Young's modulus of the matrix E_m was determined using

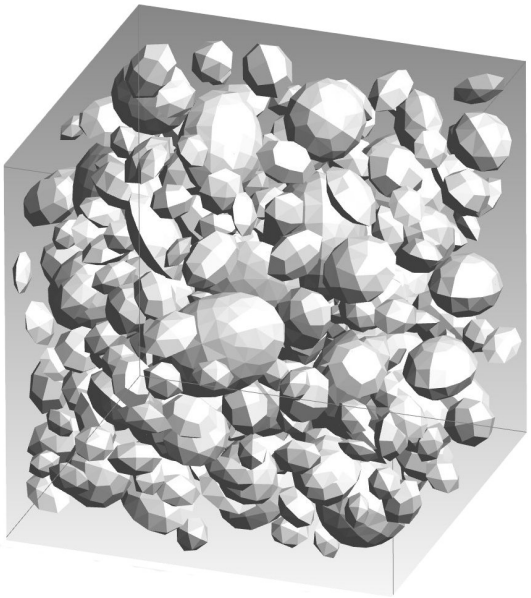


Figure 4.6 Particles simulated in the concrete cube with edge length 25mm

the Reuss formula [155]. By assuming the Young's modulus of the particles to be twice the Young's modulus of the matrix, the following relation is obtained:

$$\frac{1}{E_c} = \frac{\rho_m}{E_m} + \frac{1-\rho_m}{2E_m}, \tag{4.20}$$

where ρ_m is the volume fraction of the matrix and E_c the macroscopic Young's modulus of concrete. An alternative possibility is to fix the Young's modulus of the particles and compute the Young's modulus of the matrix.

Additionally, the influence of different yield functions used in the plasticity model has been investigated (Rankine, Drucker-Prager and combination of both) and compared to the isotropic damage model. In Fig. 4.7, the damage distribution at the end of the calculation is illustrated. One of the main motivations for using mesoscale models was the idea to represent the complex macroscopic response by discretizing the heterogenous mesostructure in the numerical model and applying simple material formulations to each of the constituents. For that matter it was assumed that the compressive failure of concrete could be modeled by using a mesoscale model with matrix failure defined by a criterion in tension as e.g. the Rankine yield surface in the combined damage/plasticity model or the isotropic damage model. The damage pattern in Fig. 4.7(b) does not coincide with the experimental results obtained for a compression test of concrete with high friction between the platens and the cube. Furthermore, no global softening behavior could be observed as illustrated in Fig. 4.8. At the ultimate applied displacement, columns of intact material transmit high compressive forces. They are separated by vertical cracks, which developed due to the lateral strain in the matrix. In a similar way, the isotropic damage model failed to represent a realistic failure pattern, although the damage distribution is quite different, as illustrated in Fig. 4.7(c). Almost the full inner zone of the cube fails, and the peak load is highly underestimated. Both formulations are based on the Rankine-criterion, but the principal difference is that, in the combined plasticity-damage

Table 4.3 Material parameters for the 3-dimensional compression test

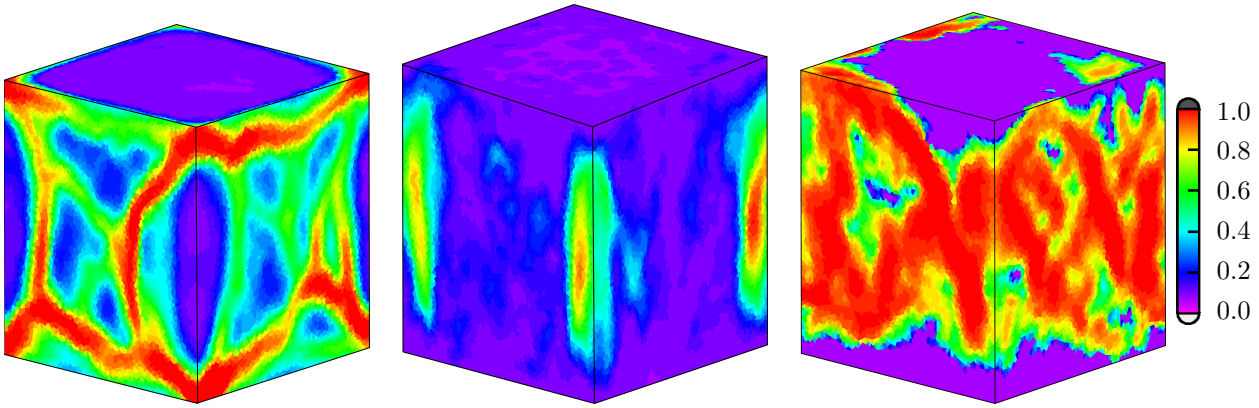
parameter			matrix	interface	aggregate
Young's modulus	E	$\left[\frac{N}{mm^2}\right]$	26738	-	53476.1
Poisson's ratio	ν	$[-]$	0.18	-	0.18
tensile strength	f_{ct}	$\left[\frac{N}{mm^2}\right]$	3.4	$f_i \cdot 3.4$	-
uniaxial compressive strength	f_c	$\left[\frac{N}{mm^2}\right]$	50	-	-
biaxial compressive strength	f_{c2}	$\left[\frac{N}{mm^2}\right]$	58	-	-
fracture energy	G	$\left[\frac{Nmm}{mm^2}\right]$	0.12	$G_i \cdot 0.12$	-
nonlocal radius	r	$[mm]$	1	-	-
interaction value	α	$[-]$	-	1	-

model due to the development of plastic strains, an anisotropic nonlinear behavior is modeled, whereas in the isotropic damage model the exceedance of the Rankine criterion in one direction automatically induces a reduction of the stiffness in all other directions. It is furthermore observed that both formulations fail to simulate the dilatancy effect as shown in Fig. 4.8(b).

An alternative formulation was based on a pure Drucker-Prager criterion, where the cone of the failure surface was determined from the uniaxial and biaxial compressive strengths. In most numerical simulations, the latter was estimated from the uniaxial compressive strength. The Drucker-Prager yield surface is often used to describe failure of concrete in compression. The failure pattern corresponds to the experimental results similar to the combined approach, where curved shear zones develop, which finally lead to spalling of the vertical edges. Additionally, softening of the global response can be simulated. The onset of failure is accurately captured by the model, as well as the lateral strain (which is calculated from the lateral displacements at the midsection). The fracture energy in compression is overestimated by the numerical model, which is probably due to the fact that the parameter ε_f is calibrated from a tensile test. Furthermore, a combination of the Rankine criterion for tension and the Drucker-Prager criterion in compression in the framework of a multiplasticity approach was investigated. The onset of material degradation was slightly earlier compared to the Drucker-Prager model. The simulated axial stress-strain diagram is close to the experimental data, although the compressive strength is slightly smaller in the numerical model, which is probably due to the parameter set (no fitting). In a similar way, the difference in the lateral strain can be explained. The volumetric expansion (dilatation) is best approximated with the combined model. The interface parameters, i.e. the factor relating the fracture energy and the tensile strength of the interface to the corresponding parameters of the matrix, have only a minor influence. This might be attributed to the small number of particles considered in the simulation. A second reason is probably the fact that in a compression test the primary failure domain is the matrix. Interface cracks seem to have only a minor influence.

4.5.3 Axisymmetric tensile test of notched specimen

The third example is a uniaxial tensile test of a prenotched cylinder experimentally investigated in [156]. The geometry of the test setup is illustrated in Fig. 4.9(a). Three LVDT's



(a) Drucker-Prager and Rankine

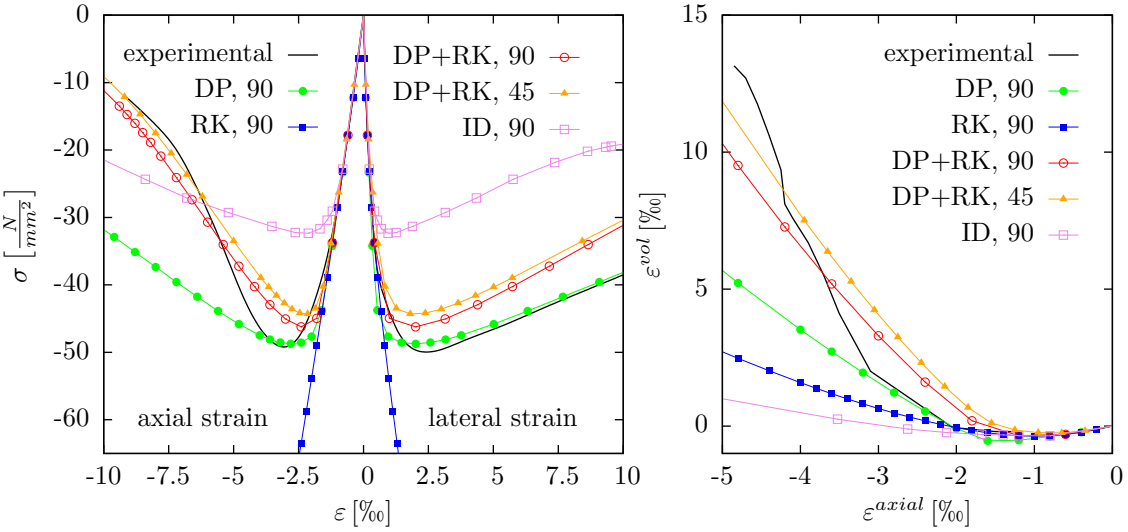
(b) Rankine yield function

(c) Isotropic damage

Figure 4.7 Damage distribution after the final load step of a uniaxial compression test for different yield functions of the matrix material using the combined damage-plasticity model and the isotropic damage model.

(linear variable differential transformer) with a gauge length of 25mm are used to measure the crack-mouth opening displacement across the weakened section, which is obtained by cutting a 10mm deep notch at the midsection using a diamond-impregnated disc. In the numerical simulation, only the central part with a length of 100mm is simulated. Due to the notch, the location of failure is prescribed, and, since the crack-mouth opening displacement is measured over a gauge length of 25mm, the influence of this reduction is considered to be negligible. An important point in this simulation is the application of an axisymmetric formulation. The geometry of the specimen as well as the loading is axisymmetric. The only problem is the mesoscale model. The assumption of an axisymmetric formulation with particles implies that each particle in the 2D model corresponds to a torus in 3D, which allows the transfer of hoop stresses. Consequently, due to the larger Young's modulus, the particles prevent the lateral contraction of the mortar matrix, which is not realistic. In order to prevent the transfer of these stresses within the particles, a linear elastic material formulation using a plane stress assumption within an axisymmetric formulation has been implemented. This allows the particles to deform in the hoop direction with vanishing stresses. Although the model still includes several unrealistic simplifications (particles are a torus, no particle placement at the axis of symmetry, elongation of particles in hoop direction), it allows for a realistic axisymmetric formulation and thus considerably reduces the computational effort compared to a full 3D model. The material parameters used in the numerical simulation are illustrated in Table 4.4. The geometry was meshed with an average element length of 0.75mm.

Aggregates are simulated using a grading curve given in [156], where two classes of grain sizes have been simulated numerically (2-5mm with a mass fraction of 15.4% and 5-15mm with a mass fraction of 40.8%). The compressive strength of the matrix was assumed to be identical to the global compressive strength measured experimentally, from which the uniaxial tensile strength and the biaxial compressive strength were estimated. The fracture energy has been approximated from the total area under the curve in the stress-crack-mouth opening displacement diagram. The parameters of the interface were obtained from the matrix

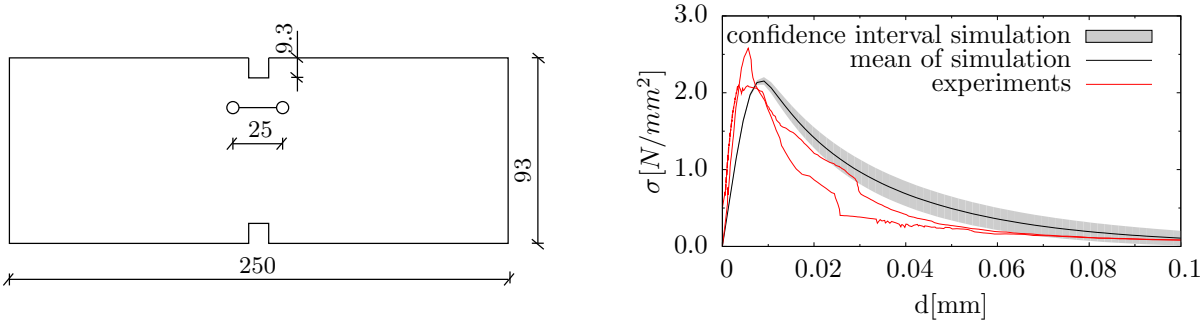


(a) Axial/lateral strain versus stress (b) Axial versus volumetric strain

Figure 4.8 Uniaxial compression test in 3D of a cube with edge length 25mm for different yield surfaces and G_i and f_i varied between 45% and 90% - (DP: combined Drucker-Prager, RK: combined Rankine, ID: Isotropic damage model).

properties with a reduction factor of 0.75 taking into account the weaker characteristics of the ITZ. By fixing the Young's modulus of the aggregates and using the rule of mixture in Eq. (4.20) with the simulated aggregate content, the Young's modulus of the matrix is obtained.

In Fig. 4.9(b), the obtained stress-crack-mouth opening displacement are plotted. The mean and the standard deviation are obtained from 100 simulations with different distributions of the particles. It is to be noted that, similar to the experiments, the stress is calculated as the total force divided by the area of the full cross-section (not the weakened one). It is observed that a discrepancy between the numerical and the experimental results already in the linear elastic response exists. For that purpose, several numerical tests including a constant Young's modulus for the full specimen, variation of the thickness and width of the cut and the



(a) Geometry (b) Stress-displacement diagram

Figure 4.9 Tensile test according to [156].

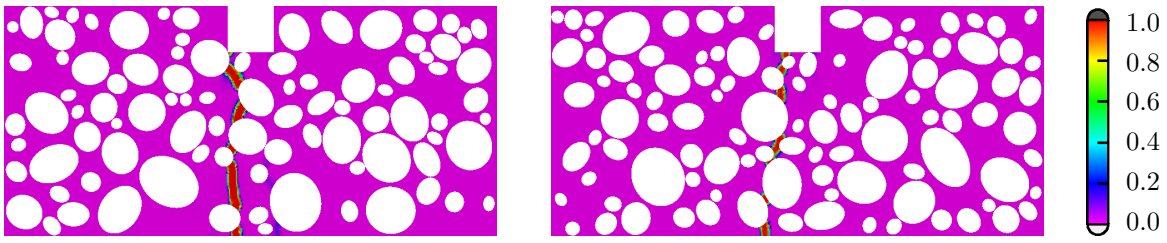
Table 4.4 Material parameters for the axisymmetric tensile test

parameter			matrix	interface	aggregate
Young's modulus	E	$\left[\frac{N}{mm^2}\right]$	≈21100	-	54000
Poisson's ratio	ν	$[-]$	0.18	-	0.2
tensile strength	f_{ct}	$\left[\frac{N}{mm^2}\right]$	4.02	3.0	-
uniaxial compressive strength	f_c	$\left[\frac{N}{mm^2}\right]$	40.2	-	-
biaxial compressive strength	f_{c2}	$\left[\frac{N}{mm^2}\right]$	46.6	-	-
fracture energy	G	$\left[\frac{Nmm}{mm^2}\right]$	0.1	0.075	-
nonlocal radius	r	$[mm]$	1.125	-	-
interaction value	α	$[-]$	-	1	-

measurement of the displacement difference at the axis of symmetry were performed. It was found that the displacement difference at the outer surface of the cylinder is approximately twice as large compared to the displacement difference at the axis of symmetry, which can be explained by the bending effects of the notch. Using a model without a notch, the linear part of the stress-crack-mouth opening displacement diagram corresponds to the measured global Young's modulus of $30 \cdot 10^3 \mathrm{N/mm^2}$. However, calculating the displacement difference in the numerical model at the axis of symmetry is not physical, since in the experiment the measurements were also performed on the outer surface. A parameter fitting procedure would be possible to match numerical and experimental curve, but since the main purpose of numerical simulations is to predict tests, where no experiments have been performed, such a fitting was not applied.

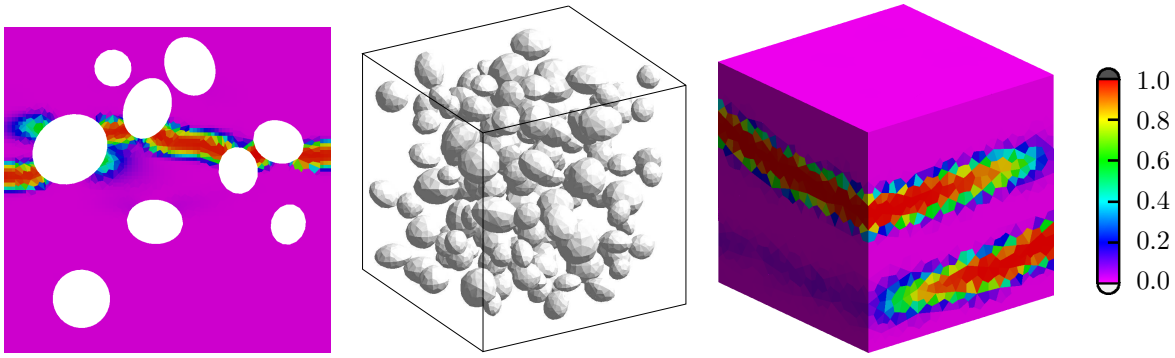
The softer response in the linear elastic part results in an overestimation of the stress in the post-peak region, since the overall fracture energy should be almost identical. However, it is to be noted that the peak load and the general shape of the stress-crack-mouth opening displacement curve can be reproduced in the numerical simulation.

In Fig. 4.10, the damage distribution at the final loading state is illustrated. It is clearly observed that the cracks tend to pass through the weak ITZ around the particles. Since a notch has been cut into the specimen, the location of the final localization zone is prescribed and the influence of the particle distribution reduces to slight variations of the crack path. Consequently, the effective crack length together with the global dissipated energy changes,



(a) sample 1 (b) sample 2

Figure 4.10 Damage distribution at the final state for two samples with a different particle distribution.



(a) Damage distribution 2D (b) Aggregates 3D (c) Damage distribution 3D

Figure 4.11 Influence of particle distribution.

which explains the increasing standard deviation in the post-peak branch of the stress-crack-mouth opening displacement diagram. For sample 1, it is furthermore observed that two competing cracks developed, but finally the damage localizes only in a single crack. With the reduced properties of the ITZ compared to the matrix material, the microcracks in the pre-peak region primarily develop in the nonlinear ITZ represented by interface elements and, consequently, almost no damage is observed outside the influencing region of the final crack.

4.5.4 Comparison 2D-3D

Most of the numerical simulations were performed under uniaxial tensile loading. This is due to the fact that the influence of the mesostructure is more pronounced under uniaxial conditions, since localization is triggered by the stochastic distribution of aggregates in the mesostructure. Furthermore, it has been realized that a compression test simulated in 2D cannot fully capture the failure mode. As a consequence, compression tests were simulated in 3D. One major disadvantage of the 3D simulations is the higher discretization effort and, especially, the higher computational cost. This is, on the one hand, due to the large number of DOF's in 3D, and, on the other hand, due to the increased band width of the stiffness matrix for the nonlocal formulation in 3D.

As a consequence, there are often two choices when modeling a tensile test. On the one hand, the tensile test could be simulated in 2D, which allows a fine discretization, the inclusion of many aggregates of different sizes (ratio between smallest and largest diameter of aggregates can be higher than 10) and straightforward visualization techniques. On the other hand, it is not clear, if the response with the 2D model can be compared to the 3D model with the same material parameters. In order to validate the results of a 2D simulation, it was tried to compare 2D and 3D simulations using a similar model with respect to the grading curve, dimension of the specimen and material parameters. Unfortunately, it was realized that with the available hardware the problem could not be solved. The numerical example is based on the Kessler-Kramer test with the details given in section 5.4. The first problem is the size of the specimen and the mesh size. A grading curve with diameters between 2mm and 8mm was used. This requires a mesh size of at least 1mm in order to be able to mesh the smallest

aggregates. A second condition gives a limit for the smallest size of the specimen: in the elastic region, the model should be an RVE and the wall effect should be reduced. Consequently, the dimension of the specimen should be at least 40x40x40mm. The 3D-model illustrated in Fig. 4.11, with a mesh size of 4mm and only aggregates in the class 4-8mm included, had already 115.000 DOF's. At first glance, this does not seem to be exceptionally large. But as already discussed before, the bandwidth of the stiffness matrix increases in 3D dramatically, especially if the nonlocal radius is larger than the mesh size. Furthermore, if the mesh size is reduced by a factor of 2, the number of DOF's increases approximately by a factor of 8. In order to be able to compare the 2D and the 3D simulation, it is further required to perform several simulations with the same material parameters in order to characterize the stochastic scatter of the response. However, using a 2D model of dimension 40x40mm with aggregates in the range 4-8mm, the number of aggregates in this model is rather small, especially if the same mesh size of 4mm is used. It might even happen, that there is no aggregate at all in the numerical model. This is due to the fact that the aggregates are simulated in 3D and a slice out of the 3D model is used, which tends to decrease the maximum size of ellipses in 2D compared to the maximum size of the ellipsoids in 3D.

As a result, it has to be concluded that mesoscale simulations with the prescribed material models in 2D and 3D are, with the current hardware, not comparable. From the 2D examples illustrated in the previous sections, it is observed that the principal failure mode corresponds to the experimentally observed pattern and it is assumed that the simplification of tensile dominated problems to 2D is not a severe restriction, whereas compressive failure is strictly to be modeled in 3D.

Chapter 5

Stochastic character of concrete on the mesoscale

5.1 Introduction

In reality, most of the processes have an uncertain character, which implies that performing an experiment or constructing a building with the same parameters twice, the response might be nonetheless different. This is due to the fact that there are certain influences that cannot be kept constant. A simple example is a compression test of similar concrete cylinders. For each specimen, a different compressive strength will be obtained, which might be due to the heterogeneity of concrete, slightly different drying conditions or compaction procedure or even different loading situations/eccentricities due to the manual insertion of the specimen into the testing machine.

In current design codes, the influence of these uncertainties is considered by the addition of safety factors for the material parameters on the one hand and the external load on the other hand, although this is only a rough approximation. Encouraged by the growing computational power and the ability to numerically simulate realistic problems, a direct consideration/simulation of these effects offers a more accurate representation of the scatter in the response.

In general, different levels of uncertainty can be distinguished. According to [157], uncertainty can be classified into stochastic, informal and lexical uncertainty. Lexical uncertainty describes the uncertainty only in a verbal way, e.g. the potential risk of failure is high, or outside it is cold, which further requires a transformation to numerical values for the representation in a numerical model. Stochastic uncertainty can be described by random variables with a prescribed density function. Sometimes it is difficult to determine the density function of a stochastic variable, e.g. if the number of samples is too small, or the stochastic influence cannot appropriately be described by a density function, which leads to the definition of informal uncertainty. An example is the determination of the compressive strength of concrete cubes in different laboratories with unknown varying boundary conditions e.g. different sizes of the test specimen, varying loading rates or temperatures.

Depending on the type of uncertainty, different models to consider these effects in a numerical model have been developed. Lexical and informal uncertainty can be simulated using the fuzzy set theory [158, 159]. In general, the uncertain variables are characterized by a membership function, which is discretized using the α-level concept [160]. In order to describe the variability of the outputs, the concept of α-level optimization can be used [161].

In the case of stochastic uncertainty, it is important to identify the sources of uncertain behavior and appropriately represent the identified phenomena in the corresponding model.

This requires, in a first step, the extraction of the relevant stochastic parameters and an experimental determination of their scatter. From these data, the stochastic model can be derived, e.g. by describing the uncertainty with random variables and their distribution and correlations or by using a random field. In a final step, the influence of these random variables on the system can be evaluated in the numerical simulation.

A generalization of the stochastic uncertainty and the fuzzy approach is the concept of fuzzy randomness [162]. In this context, the uncertain variables are characterized by an uncertain stochastic distribution, which is described by fuzzy parameters. As a consequence, the uncertainty with respect to the parameters of the probability distribution can be investigated. A disadvantage of this approach is the additional computational effort, because in combination with the α-level optimization concept for each α-level two optimization problems (min and max) have to be solved, where the objective function is calculated from a stochastic simulation, which itself is often already numerically expensive. An alternative approach describes the uncertain stochastic parameters by stochastic variables [163].

This work focuses on the investigation of the influence of stochastic random material parameters in the mesoscale model on the system response. It is further assumed that the external conditions (loading, geometry) remain constant, although in reality, they have a stochastic character.

5.2 Random variables

5.2.1 Definition

A continuous random variable X describes the stochastic behavior of a certain parameter and is characterized by the cumulative distribution function (cdf) F_X:

$$F_X(x) = P(X < x), \tag{5.1}$$

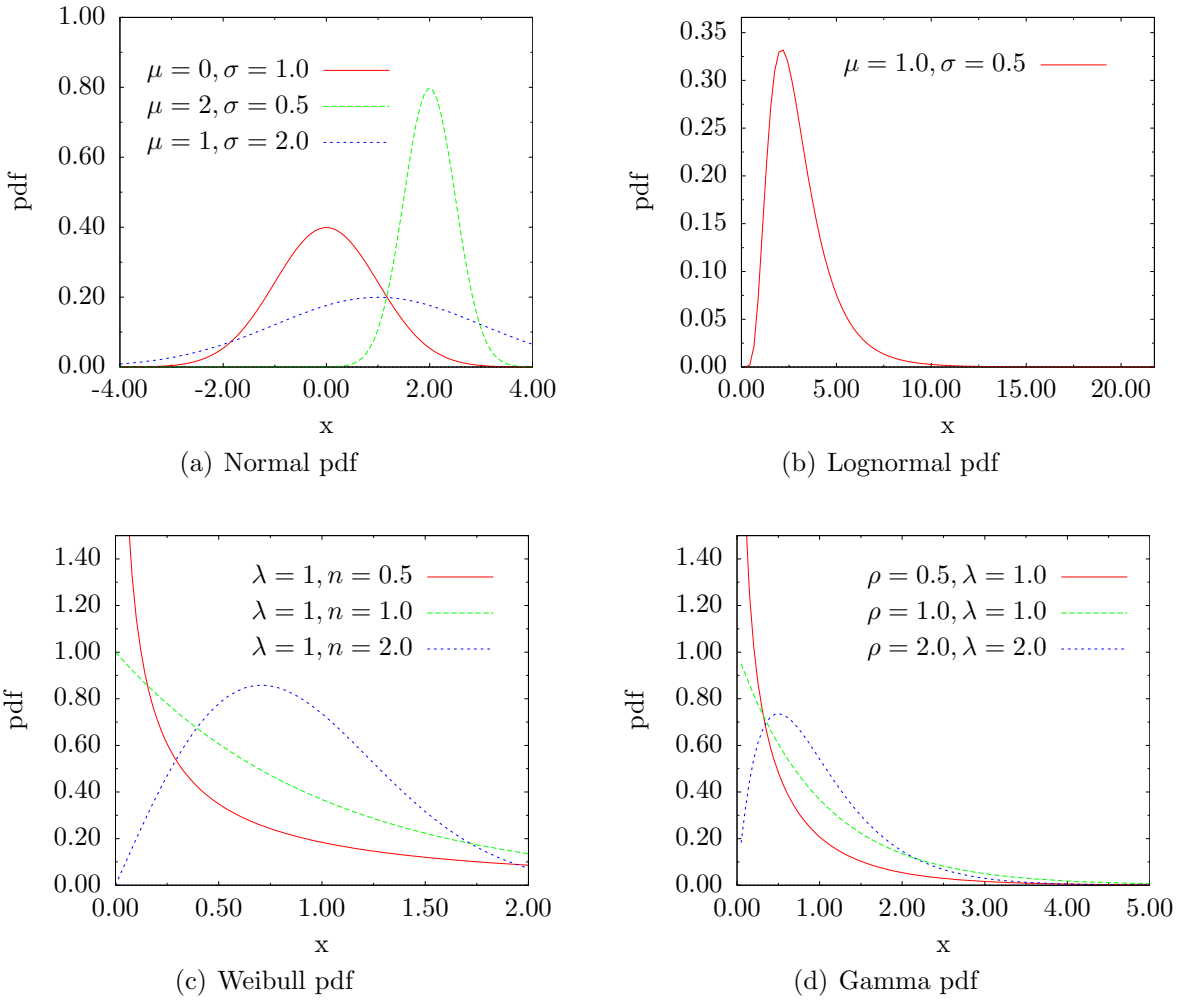
where $P(X < x)$ is the probability for $X < x$. The function $f(x) = \dfrac{\partial F_X(x)}{\partial x}$ is the probability density function (pdf) of a random variable. In Fig. 5.1, different distribution types of random variables are illustrated. The probability density function of a random variable can be described by a limited number of parameters - generalized moments of a random variable:

$$\text{mean value :} \qquad \mu = E(x) = \int_{-\infty}^{\infty} x f(x) dx \tag{5.2}$$

$$\text{variance :} \qquad \sigma^2 = E\left((x-\mu)^2\right) = \int_{-\infty}^{\infty} (x-\mu)^2 f(x) dx \tag{5.3}$$

$$\text{skewness :} \qquad \gamma_1 = \frac{1}{\sigma^3} E\left((x-\mu)^3\right) = \frac{1}{\sigma^3} \int_{-\infty}^{\infty} (x-\mu)^3 f(x) dx \tag{5.4}$$

$$\text{kurtosis :} \qquad \gamma_2 = \frac{1}{\sigma^4} E\left((x-\mu)^4\right) = \frac{1}{\sigma^4} \int_{-\infty}^{\infty} (x-\mu)^4 f(x) dx. \tag{5.5}$$



(a) Normal pdf (b) Lognormal pdf

(c) Weibull pdf (d) Gamma pdf

Figure 5.1 Different distribution types of random variables.

The mean or expected value of a distribution characterizes the average value of the random variable. For a large number of samples, the mean value can be approximated by the arithmetic average of the samples. The variance of a distribution describes the scatter of the random variable, i.e. the expected value of the deviation from the mean value. The mean value and the standard deviation are sufficient to unambiguously describe the free parameters of a normal distribution. The skewness of a random variable characterizes its asymmetry and the kurtosis describes the flatness of a distribution.

5.2.2 Sampling of univariate random variables

In the following it is assumed that uniformly distributed random numbers in the interval $[0, 1]$ can be simulated. In the current implementation, the software Mersenne Twister by [164] is used for this purpose, which generates, starting from a set of initial values, a quasi-random sequence (period of $2^{19937} - 1 \approx 4,3 \cdot 10^{6001}$) of a uniformly distributed random variable. This approach has the advantage that, although the simulation is random, it can exactly be

Table 5.1 Probability density function (pdf), mean value and standard deviation for common distribution types

distribution	pdf	mean μ	standard deviation σ	interval with nonzero pdf
Uniform	$\frac{1}{b-a}$	$\frac{a+b}{2}$	$\frac{b-a}{\sqrt{12}}$	$[a,b]$
Normal	$\frac{1}{\sigma\sqrt{2\pi}}\boldsymbol{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}$	μ	σ	$[-\infty,\infty]$
Lognormal	$\frac{1}{x\xi\sqrt{2\pi}}\boldsymbol{e}^{-\frac{(\ln x-\lambda)^2}{2\xi^2}}$	$\boldsymbol{e}^{\lambda+\frac{\xi^2}{2}}$	$\mu^2\left(\boldsymbol{e}^{\xi^2}-1\right)$	$[\lambda,\infty]$
Weibull	$\frac{n}{\lambda}\left[\frac{x}{\lambda}\right]^{n-1}e^{-\left(\frac{x}{\lambda}\right)^n}$	$\lambda\Gamma\left(1+\frac{1}{n}\right)$	$\sqrt{\lambda^2\left[\Gamma(1+\frac{2}{n})-\Gamma^2\left(1+\frac{1}{n}\right)\right]}$	$[0,\infty]$
Gamma	$x^{\rho-1}\frac{e^{-\frac{x}{\lambda}}}{\Gamma(\rho)\lambda^\rho}$	$\rho\lambda$	$\rho\lambda^2$	$[0,\infty]$

reproduced when using the same initial seeds. Different approaches to simulate are random variable can be found in the literature, see e.g. [165] for a short review.

5.2.2.1 Inversion method

The first method is the inversion method or the method of transformation of variables. The basic idea is illustrated in Fig. 5.2. For a cumulated probability density function F with inverse F^{-1} defined by

$$F^{-1}(u) = \sup\left\{x : F(x) < u, \text{with}\, 0 \leq u \leq 1\right\}, \tag{5.6}$$

and U a uniform random variable in the interval $[0,1]$, the variable $X = F^{-1}(U)$ has the cumulated probability density function F. If the inverse of the distribution function cannot be derived analytically, a numerical procedure as e.g. a Newton iteration can be applied. For the standard normal distribution, the representation of the normal cdf using the error function $\text{erf}(x)$ can be used to determine the inverse cumulated distribution function:

$$F(x) = \frac{1}{\sqrt{2\pi}}\int_{u=-\infty}^{x} \mathrm{e}^{-\frac{u^2}{2}}\, du = \frac{1}{2}\left[1+\text{erf}\left(\frac{x}{\sqrt{2}}\right)\right] \tag{5.7}$$

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}}\int_{u=0}^{x} e^{-u^2}\, du \tag{5.8}$$

where the inverse of the error function can be approximated e.g. by [166].

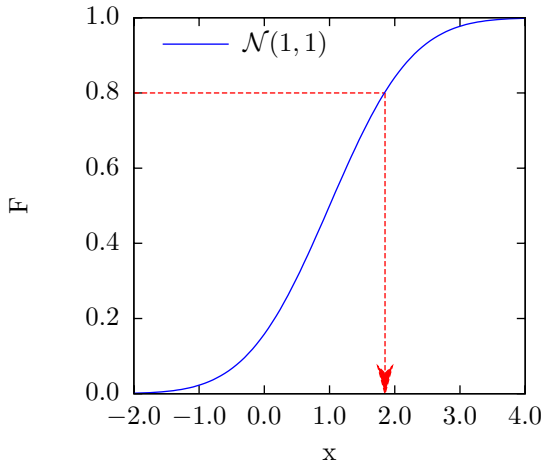


Figure 5.2 Inversion method for sampling a random variable from a uniform distribution.

5.2.2.2 Rejection sampling

A second possibility is rejection sampling. Assuming that a simple probability density function (pdf) Q is given, from which samples can be generated. It is further assumed that a constant c exists, with

$$cQ(x) > P(x), \tag{5.9}$$

where P is the pdf from which samples should be generated. Two random numbers are generated as illustrated in Fig. 5.3. The first one, x, from the pdf Q and a second random variable u uniformly distributed in the interval $[0, cQ(x)]$. If $u < P(x)$, the sample is accepted, otherwise it is rejected. The procedure is repeated until all samples x are generated. The method works good, if P and Q are similar, otherwise the constant c is large and the rejection rate is high.

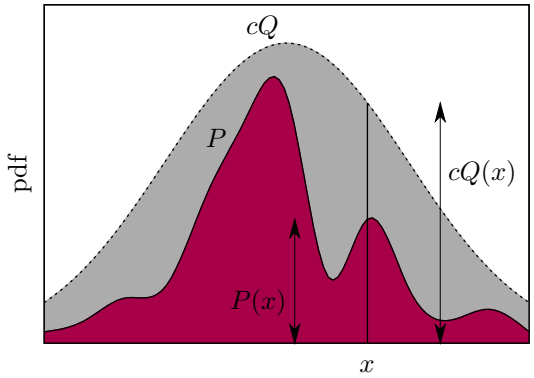


Figure 5.3 Rejection method for sampling a random variable from a uniform distribution.

5.2.2.3 Special cases

For certain distribution types, special algorithms to simulate random numbers have been developed, primarily with the aim of speeding up the calculation of many random numbers. A fast algorithm for the normal and the exponential distribution are the Ziggurat method [167], which is a fast rejection method that uses precalculated values in order to replace the number of expensive calculations of the density function by simple comparisons of real numbers, and a fast random number generator by [168], which determines random variables by a transformation using already generated random variables. For the normal distribution, a sample can be obtained using the Box-Muller transform [169] or an extension using the polar form. An extensive overview is given in [170] and a comparison of the speed and accuracy of different methods for gaussian random variables is given in [171].

5.2.3 Sampling of multivariate random variables

In the previous section, different methods to sample single variates according to a prescribed distribution have been discussed. A practical problem is often described by many random variables. If they are uncorrelated, they can be sampled one at a time. However, often these variables are correlated, e.g. the Young's modulus and the tensile strength of concrete. The correlation between variables can be described by their covariance

$$\mathrm{COV}(X_i, X_j) = E\left[(X_i - \bar{X}_i)(X_j - \bar{X}_j)\right], \tag{5.10}$$

where $\bar{X}_i$ is the mean value. Variables which have a covariance of zero are called uncorrelated. A dimensionless measure of the linear correlation is defined by the coefficient of correlation (Pearson's correlation):

$$\rho_{ij} = \frac{E\left[(X_i - \bar{X}_i)(X_j - \bar{X}_j)\right]}{\sigma_i \sigma_j} = \frac{\mathrm{COV}(X_i, X_j)}{\sigma_i \sigma_j}. \tag{5.11}$$

It is to be noted that the coefficient of correlation and the covariance are zero for independent variables. The inverse is not necessarily true, e.g. consider samples uniformly distributed on a circle. Their correlation is zero, whereas they are far from independent. Under the assumption of normal distributed variables, uncorrelated implies also independence. The coefficient of correlation, which is in the range between -1 and 1 is furthermore only valid for distributions that are close to the normal distribution. Consider e.g. two random variables which have a correlation coefficient of almost one. This implies that their pdf is identical, apart from the scaling with the standard deviation and the translation by the mean value.

Other nonparametric measures of correlation are e.g. Spearman's ρ [172] or Kendall's τ [173], which are based on rank correlation. The principal idea is that the correlation is not determined between values x_1 and x_2 of two random variables, but between their rank in a set of samples. These rank based measures have the advantage that no underlying assumption with respect to the distribution are required, but a unique description of the actual dependence is only obtained under certain restrictions, e.g. a multivariate normal distribution.

In order to simulate correlated random variables, different procedures have been investigated. The first strategy is based on the Nataf model, which assumes a jointly normal pdf. For normally distributed variables, this procedure is exact and accurate approximations within certain limits for the correlation coefficient can be obtained for non-gaussian variates. The second model is based on the Metropolis-Hastings algorithm, which yields for any joint probability density function a corresponding set of samples. However, the correlation between these samples requires more samples to be generated than are actually required.

5.2.3.1 Monte-Carlo simulation using the Nataf model

The Nataf model was developed by [174] based on [175] to simulate correlated random variables. Given random variables $\boldsymbol{X} = (X_1, .., X_n)$, a transformation to standard gaussian random variables $\boldsymbol{Y} = (Y_1, .., Y_n)$ can be performed by

$$Y_i = \Phi^{-1}\left[F_{X_i}\left(X_i\right)\right], \tag{5.12}$$

where Φ is the standard cumulative normal probability. The probability density function of the original variables can be expressed as a function of the transformed variables by

$$f_X(\boldsymbol{x}) = f_Y(\boldsymbol{y}) \left| \frac{d\boldsymbol{y}}{d\boldsymbol{x}} \right|, \tag{5.13}$$

where $|\ \ |$ describes the determinant of a matrix. Using the assumption of [175] that Y is jointly normal with covariance $\boldsymbol{\Sigma}^*$ ($f_y(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}^*)$), the transformation in Eq. (5.12) gives

$$f_X(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}, \boldsymbol{\Sigma}^*) \prod_{i=1}^{N} \frac{dy_i}{dx_i} = \mathcal{N}(\boldsymbol{y}, \boldsymbol{\Sigma}^*) \prod_{i=1}^{N} \frac{f_{X_i}(x_i)}{\phi(y_i)}, \tag{5.14}$$

where ϕ is the pdf of a standard gaussian variable and

$$\frac{dy_i}{dx_i} = \frac{1}{\Phi'\underbrace{\left[\Phi^{-1}(F_{X_i}(x_i))\right]}_{y_i}} \frac{dF_{X_i}(x_i)}{dx_i} = \frac{f_{X_i}(x_i)}{\phi(y_i)}, \tag{5.15}$$

where the quotient and the chain rules are used with Φ' being the derivative of the cumulative standard normal distribution function. Furthermore, the derivative of the inverse function is identical to the inverse of the derivative of the original function:

$$\left(\Phi^{-1}(x)\right)' = \frac{1}{\Phi'(x)}. \tag{5.16}$$

The elements of the correlation matrix $\boldsymbol{C}^*$ are related to the correlation matrix $\boldsymbol{C}$ of the original distribution with coefficients ρ_{ij} by

$$\rho_{ij} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left(\frac{x_i - \mu_i}{\sigma_i}\right)\left(\frac{x_j - \mu_j}{\sigma_j}\right) f_{X_i}(x_i) f_{X_j}(x_j) \frac{\phi_2\left(y_i, y_j, \rho_{ij}^*\right)}{\phi(y_i)\phi(y_j)}\, dx_i dx_j \tag{5.17}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left(\frac{x_i - \mu_i}{\sigma_i}\right)\left(\frac{x_j - \mu_j}{\sigma_j}\right) \phi_2\left(y_i, y_j, \rho_{ij}^*\right)\, dx_i dx_j, \tag{5.18}$$

Table 5.2 Correlation coefficients $\boldsymbol{C}^* = (\rho_{ij}^*)$ as a function of $\boldsymbol{C} = (\rho_{ij})$ for the Nataf model and maximum error in % according to [174] with δ_{jj} coefficient of variation, N(normal distribution), U(uniform distribution), LN(lognormal distribution)

X_j \ X_i	N	U	LN
N	ρ_{ij}(exact)		
U	$1.023\rho_{ij}$(0%)	$1.047 - 0.047\rho_{ij}^2$(0.0%)	
LN	$\frac{\delta_{jj}}{\sqrt{\ln(1+\delta_{jj}^2)}}$ (exact)	$1.019 + 0.014\delta_{jj} + 0.010\rho_{ij}^2 + 0.249\delta_{jj}^2$ (0.7%)	$\frac{\ln(1+\rho_{ij}\delta_{ii}\delta_{jj})}{\rho_{ij}\sqrt{\ln(1+\delta_{ii}^2)\ln(1+\delta_{jj}^2)}}$ (exact)

where Eq. (5.15) has been used and ϕ_2 is the density function of two standard normal distributed variables. The preceding transformation is valid as long as $\boldsymbol{C}^*$ is positive definite and the cumulated probability density functions F_{X_i} are strictly increasing. The first condition is, for most practical problems, fulfilled since $\boldsymbol{C}$ is by definition positive definite and the differences between $\boldsymbol{C}$ and $\boldsymbol{C}^*$ are small. For each pair of marginal distributions, Eq. (5.18) can be solved iteratively for ρ_{ij}^*. In order to avoid the recalculation of the integrals, [174] provided regression formulas for certain pdf's and their limits. An extract of the derived relations is given in Table 5.2. Once the correlation in the correlated Gaussian space is obtained, the transformation into the uncorrelated Gaussian space is required, which is performed by a lower triangular matrix $\boldsymbol{L}$ obtained from the Cholesky decomposition of the covariance matrix $\boldsymbol{\Sigma}^*$:

$$\boldsymbol{\Sigma}^* = \boldsymbol{L}\boldsymbol{L}^T \tag{5.19}$$

$$\boldsymbol{Y} = \boldsymbol{L}\boldsymbol{Z}. \tag{5.20}$$

The pdf of the transformed variables $\boldsymbol{Z}$ can be expressed as

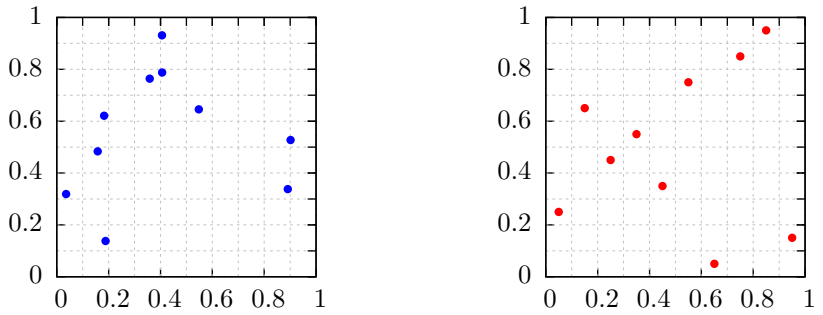
$$f_Z(\boldsymbol{z}) = f_Y(\boldsymbol{y})\left|\frac{d\boldsymbol{y}}{d\boldsymbol{z}}\right| = f_Y(\boldsymbol{y})\,|\boldsymbol{L}| = \frac{|\boldsymbol{L}|}{(2\pi)^{N/2}\sqrt{|\boldsymbol{\Sigma}^*|}}\boldsymbol{e}^{-\frac{1}{2}\boldsymbol{y}^T(\boldsymbol{\Sigma}^*)^{-1}\boldsymbol{y}} \tag{5.21}$$

$$= \frac{|\boldsymbol{L}|}{(2\pi)^{N/2}\sqrt{|\boldsymbol{L}||\boldsymbol{L}^T|}}\boldsymbol{e}^{-\frac{1}{2}\boldsymbol{z}^T\boldsymbol{L}^T\left(\boldsymbol{L}\boldsymbol{L}^T\right)^{-1}\boldsymbol{L}\boldsymbol{z}} = \frac{1}{(2\pi)^{N/2}}\boldsymbol{e}^{-\frac{1}{2}\boldsymbol{z}^T\boldsymbol{z}} \tag{5.22}$$

$$= \prod_{i=1}^{N}\frac{1}{\sqrt{2\pi}}\boldsymbol{e}^{-\frac{1}{2}z_i^2} \tag{5.23}$$

where the relation $(\boldsymbol{A}\boldsymbol{B})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1}$ has been used. The standard normal distributed random variables $\boldsymbol{Z}$ are uncorrelated and can be expressed as a product of single variate density functions.

In summary, the generation of samples $\boldsymbol{x}$ using the Nataf model is performed by generation of a sample $\boldsymbol{z}$ in the uncorrelated Gaussian space with $\mathcal{N}(0,1)$, transformation in the correlated



(a) Monte Carlo simulation (b) Latin hypercube simulation

Figure 5.4 Comparison of crude Monte Carlo simulation and Latin Hypercube sampling for two random variables uniformly distributed in the interval [0,1] with 10 samples.

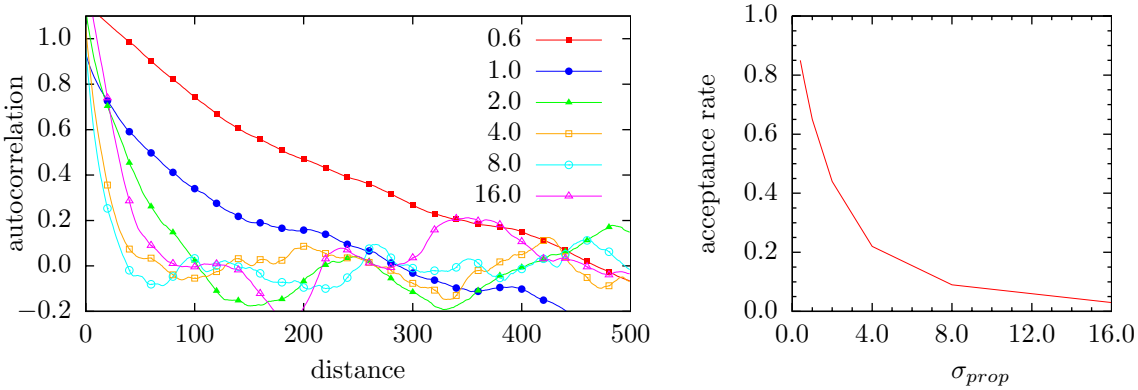
Gaussian space according to $\boldsymbol{y} = \boldsymbol{L}\boldsymbol{z}$ and a transformation to the prescribed distribution according to $x_i = F_{X_i}^{-1}\left[\Phi\left(\boldsymbol{y}\right)\right]$. The calculation of the Cholesky decomposition $\boldsymbol{L}\boldsymbol{L}^T$ can be performed in advance for all samples.

5.2.3.2 Latin-Hypercube sampling

Latin-Hypercube sampling, originally proposed by [176], is a sampling strategy that often requires less samples compared to standard Monte-Carlo simulations to achieve the same accuracy. The basic idea is to divide the domain of each variable into disjunct equiprobable intervals. A representative point in each of the intervals is chosen, which might be sampled using crude Monte-Carlo in each interval as in the original version, taking the center point of the interval [177, 178], which is restricted to finite intervals, or using the mean of each interval [179]. In a second step, these representative points for each variable are ordered randomly in a matrix $\boldsymbol{R}_{NM}$, where N is the number of samples and M the dimension of the problem. The columns of $\boldsymbol{R}$ represent a random permutation of the rank numbers between 1 and N. Consequently, the i^{th} sample is obtained by using for each variable the i^{th} representative point, which is related to the i^{th} row of $\boldsymbol{R}$. The results of a sampling are illustrated in Fig. 5.4. It is observed that due to the stratification of the random variables the probability density function of a single variable is better represented using a limited number of samples. Using this procedure, this might introduce significant correlations between the variables in the samples.

In order to avoid these artificial correlations, an updated Latin Hypercube sampling has been used [180, 181]. The idea is to describe the correlation by the Spearman correlation coefficient, which defines correlations between the rank numbers of samples. These rank numbers are given by the corresponding rows of $\boldsymbol{R}$. The Spearman correlation coefficient for a sample set can be calculated from

$$T_{ij} = 1 - \frac{6\sum_{k=1}^{N}\left(R_{ki} - R_{kj}\right)^2}{N(N^2-1)}. \tag{5.24}$$



(a) Autocorrelation for X_1 using different $\sigma_{prop} = 0.6 - 16$ (b) Acceptance rate

Figure 5.5 Autocorrelation for the Markov chain and acceptance rate in the sampling of two normal distributed random variables with $\boldsymbol{\mu} = (4, 4)$, $\sigma = (3, 3)$ and a correlation of 0.9.

A Cholesky decomposition of the matrix $\boldsymbol{T} = \boldsymbol{Q}\boldsymbol{Q}^T$ can be performed, since the matrix is symmetric and positive definite as long as all variables have a different ordering scheme. Afterwards, an auxiliary ordering matrix $\bar{\boldsymbol{R}}$ is defined by

$$\bar{\boldsymbol{R}} = \boldsymbol{R}\boldsymbol{Q}^{-1} \tag{5.25}$$

and each column of the permutation matrix $\boldsymbol{R}$ is reordered according to the ranks of the corresponding column in the auxiliary matrix $\bar{\boldsymbol{R}}$. This process can be repeated several times (in the current implementation it is repeated 10 times) to sample uncorrelated variables. Standard normal distributed samples are obtained from the ranks by

$$z_i^{(k)} = \Phi^{-1}\left(\frac{R_{ki} - 0.5}{N}\right), \tag{5.26}$$

where Φ^{-1} is the inverse cumulative distribution function of the normal distribution and N is the number of samples. In order to obtain correlated random variables, the Nataf-model as illustrated in the previous section is used, with the difference that the uncorrelated Gaussian random variables are samples with a latin hypercube sampling.

5.2.3.3 Metropolis-Hastings algorithm

A method developed by [182] and generalized in [183] is the Metropolis-Hastings algorithm, which shares some features with the rejection method. Again, a simple pdf Q is given, from which samples can be generated. In contrast to the rejection method, the pdf may depend on the position $\boldsymbol{x}$, in which case it is called random walk Metropolis-Hastings algorithm. Starting from a point $\boldsymbol{x}^{(k)}$ in the Markov chain (which is a stochastic process, where the future states depend only on the current state, and not on the past states), a next point $\bar{\boldsymbol{x}}$ is generated from Q, which can be e.g. a multivariate normal distribution $\mathcal{N}(\boldsymbol{x}^{(k)}, \boldsymbol{\Sigma}_{prop})$ centered at $\boldsymbol{x}^{(k)}$.

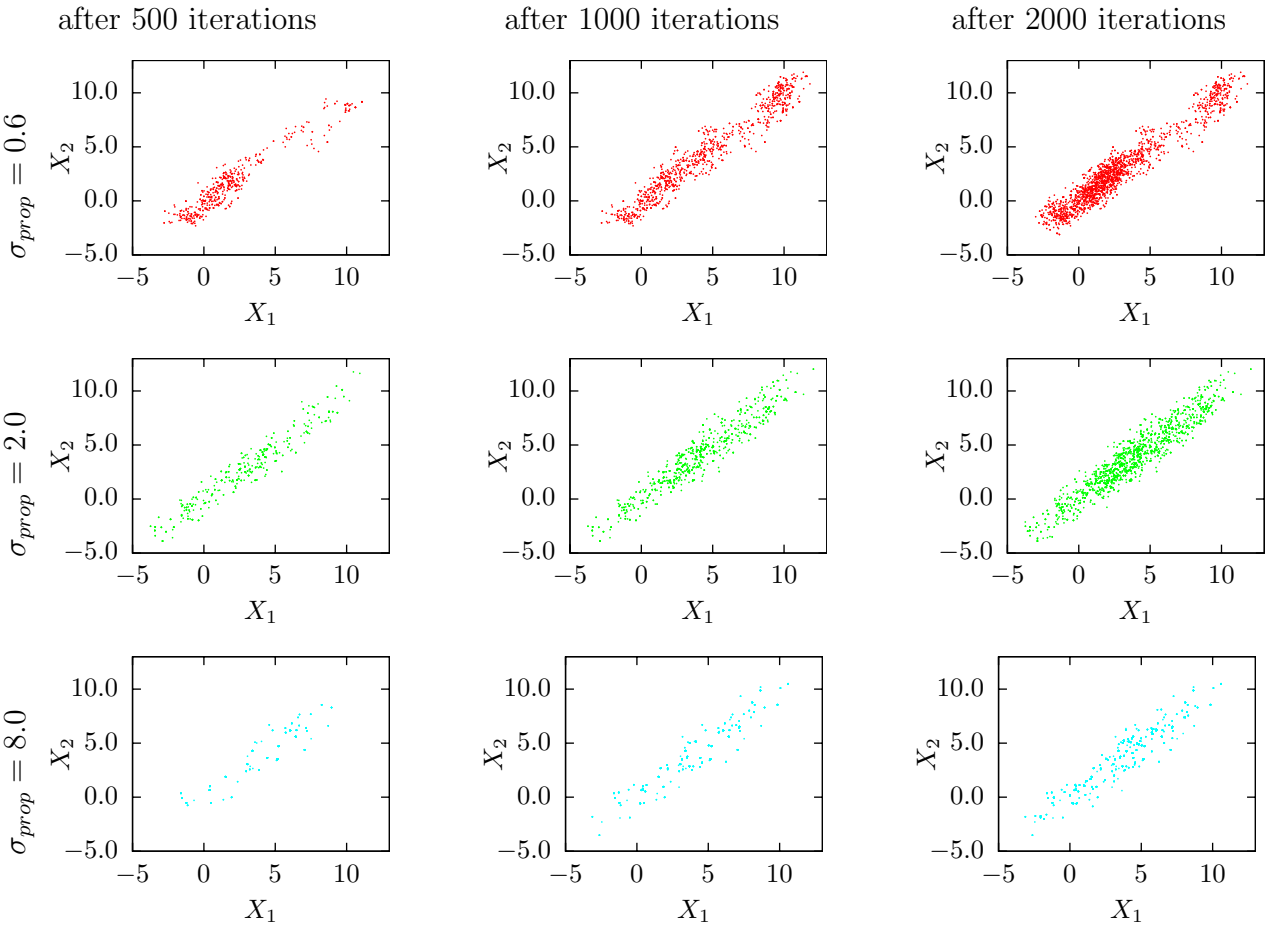


Figure 5.6 Iterative generation of samples using the Metropolis-Hastings algorithm with different standard deviations of the sampling density σ_{prop}.

The sample is accepted ($\boldsymbol{x}^{(k+1)} = \bar{\boldsymbol{x}}$), if a second random variable α drawn from the uniform distribution in $[0, 1]$ fulfills $\alpha < a$ with

$$\boldsymbol{x}^{(k+1)} = \begin{cases} \bar{\boldsymbol{x}} & \alpha < a \\ \boldsymbol{x}^{(k)} & \text{otherwise} \end{cases} \qquad \text{with} \qquad a = \frac{P(\bar{\boldsymbol{x}})}{P(\bar{\boldsymbol{x}}^{(k)})} \frac{Q(\boldsymbol{x}^{(k)}|\bar{\boldsymbol{x}})}{Q(\bar{\boldsymbol{x}}|\boldsymbol{x}^{(k)})}. \tag{5.27}$$

The pdf $Q(\boldsymbol{x}|y)$ is the probability function at x of a proposal density Q depending on y, e.g. $\mathcal{N}(y, \sigma I_N)$. If the point is rejected, the previous point of the Markov chain is copied, which is in contrast to rejection sampling, where rejected points have no influence on the list of samples. The difficulty of the method is the determination of the covariance $\boldsymbol{\Sigma}_{prop}$, or, under the assumption of an uncorrelated sampling density function with identical variance for all variables, the variance σ_{prop} of the pdf Q. If σ_{prop} is chosen too small, successive samples in the Markov chain are highly correlated and only small steps are performed, which results in a large number of iterations required to obtain an ergodic process. An example of two normal distributed variables with $\boldsymbol{\mu} = (4, 4)$, $\sigma = (3, 3)$ and a correlation of 0.9 demonstrates the

influence of the standard deviation of the sampling density function σ_{prop}. The autocorrelation in the Markov chain, illustrated in Fig. 5.5(a) for the first variable X_1, is calculated from

$$R(k) = \frac{1}{(n-k)\,\sigma^2} \sum_{i=1}^{n-k} \left(x_1^{(i)} - \mu_1 \right) \left(x_1^{(i+k)} - \mu_1 \right), \tag{5.28}$$

where the exact values for the parameters μ and σ have been used to obtain an unbiased estimate of the autocorrelation. For a limited number of samples, this approach might give correlations larger than one. The correlation length has a minimum at about $\sigma_{prop} = 8$. For higher values, an increase of the correlation length, which corresponds to the value k at which the autocorrelation is almost zero, is obtained. This is due to the fact that for large σ_{prop} the acceptance rate is small which is illustrated in Fig. 5.5(b). Consequently, successive samples are often identical. For smaller σ_{prop}, the step length between successive samples is too small which similarly generates an artificial correlation. Usually, the parameter σ_{prop} is determined from the acceptance rate, which should be between 0.2 and 0.5. In order to obtain independent samples, the correlation length can be determined and only every k^{th} sample of the chain is used.

Another important point is the influence of the starting point. After a certain period, which is related to the correlation length, the point at $\boldsymbol{x}^{(k)}$ is independent from the starting point $\boldsymbol{x}^{(0)}$.

In Fig. 5.6, the generation process of samples for a chain with 5000 elements, where the first 3000 have been omitted, is illustrated. It is demonstrated that for small σ_{prop} the samples are concentrated in certain parts of the domain, which is due to the small step size between consecutive samples and the high correlation. For large σ_{prop}, the number of identical samples is high, e.g. for $\sigma_{prop} = 8.0$ an acceptance rate of 0.09 is obtained, which implies that, on average, every sample is ten times in the set.

5.3 Random fields

Correlations between different material parameters at the same material point can be described by using correlated random variables. In order to describe the spatial correlation between neighboring material points, random fields can be used. A random field is a multi-dimensional stochastic process $H(\boldsymbol{x})$, where $\boldsymbol{x} \in \Omega \subseteq \mathcal{R}^n$, where n is the dimension of the random field.

In a first part, the algorithm is develop for a univariate random field and, afterwards, extended to multiple variates.

5.3.1 Univariate random fields

A univariate random field is characterized by its distribution (e.g. normal distributed with given mean value and standard deviation). If the distribution is independent of the location $\boldsymbol{x}$, the random field is called homogeneous.

An important characteristic of a random field is its spatial correlation. This can be described by the covariance function Σ_{uu}, which is given for a homogeneous random field u by:

$$\Sigma_{uu}(\boldsymbol{\xi}) = E[u(\boldsymbol{x})u(\boldsymbol{x}+\boldsymbol{\xi})] - E[u(\boldsymbol{x})]^2. \tag{5.29}$$

The corresponding non-dimensional quantity is the correlation function

$$C_{uu}(\boldsymbol{\xi}) = \frac{\Sigma_{uu}(\boldsymbol{\xi})}{\sigma_u^2} \tag{5.30}$$

with σ_u^2 being the variance of the random field. In a similar way, the spatial correlation can be described by the auto-correlation function given by

$$R_{uu}(\boldsymbol{\xi}) = E[u(\boldsymbol{x})u(\boldsymbol{x}+\boldsymbol{\xi})] = \Sigma_{uu}(\boldsymbol{\xi}) + E[u(\boldsymbol{x})]^2, \tag{5.31}$$

where $\boldsymbol{\xi}$ is the distance between a point $\boldsymbol{x}$ and another point $\boldsymbol{x}+\boldsymbol{\xi}$.

A common auto-correlation function is described by an exponential decay

$$R_{uu}(\boldsymbol{\xi}) = \sigma_u^2 \exp\left(-\frac{|\boldsymbol{\xi}|^2}{l_h^2}\right), \tag{5.32}$$

where l_h is the correlation length. It is to be noted that Eq. (5.32) corresponds to an isotropic random field, i.e. where only the distance between two points - and not the direction - influences their correlation. If the correlation length tends to zero, no spatial correlation between neighboring points is obtained, whereas for $l_h \to \infty$ all values of the random field within Ω are identical, which allows the description of the stochastic random field by a single random variable.

5.3.2 Multivariate random fields

The spatial variability of material properties within a structure is assumed to be influenced by the same phenomena (e.g. different mixture, pore density). This leads to the assumption that the shape of the auto-correlation function is identical for all properties, i.e. same auto-correlation function and the same correlation length, which gives, similar to Eq. (5.32), the cross-correlation function between material properties i and j as e.g. the Young's modulus E and the tensile strength f_{ct}

$$R_{ij}(\boldsymbol{\xi}) = \rho_{ij}\sigma_i\sigma_j \exp\left(-\frac{|\boldsymbol{\xi}|}{l_h}\right), \tag{5.33}$$

where ρ_{ij} is the correlation coefficient between material properties i and j. For $i = j$, this reduces to the autocorrelation function in Eq. (5.32).

5.3.3 Overview of discretization techniques

In order to consider the influence of the spatial variability of material parameters in a numerical simulation, the random field has to be discretized at certain points. In the context of the finite element method, different approaches have been developed. Material properties are evaluated on the integration point level. As a consequence, it seems to be natural to discretize the random field on the integration point level, which results in one random variable per integration point [184]. This requires the element size to be small compared to the correlation length. The covariance between these random variables and the correlation can be derived from the autocorrelation function by

$$\boldsymbol{\Sigma}_{uu}(\boldsymbol{U}_i, \boldsymbol{U}_j) = R_{uu}(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|) - E[u(\boldsymbol{x})]^2 \tag{5.34}$$

$$\boldsymbol{C}_{uu}(\boldsymbol{U}_i, \boldsymbol{U}_j) = \frac{1}{\sigma_u^2}\boldsymbol{\Sigma}_{uu}(\boldsymbol{U}_i, \boldsymbol{U}_j), \tag{5.35}$$

where $\boldsymbol{x}_i$ are the integration point coordinates, U_i is the random variable at $\boldsymbol{x}_i$ and $E[u(\boldsymbol{x})]$ is the mean value of the random field. A sampling of these correlated random variables can then be performed with the method described for correlated random variables in section 5.2.3.1. It is to be emphasized that, for a large finite element model, the Cholesky decomposition of the covariance matrix in Eq. (5.19) is the limiting factor. Depending on the correlation length, the covariance matrix $\boldsymbol{C}_{uu}$ is, in general, not sparse and, consequently, the Cholesky decomposition of a full matrix with the dimension corresponding to the number of integration points has to be calculated.

An alternative, especially suited for a large correlation length, is an eigenvalue decomposition of the covariance matrix instead of a Cholesky decomposition:

$$\boldsymbol{C}_{uu} = \boldsymbol{\Phi}_n^T \mathrm{diag}(\lambda_1, \lambda_2, .., \lambda_n)\boldsymbol{\Phi}_n \tag{5.36}$$

$$\approx \boldsymbol{\Phi}_k^T \mathrm{diag}(\lambda_1, \lambda_2, .., \lambda_k, 0, .., 0)\boldsymbol{\Phi}_k, \tag{5.37}$$

where $\boldsymbol{\Phi}_k$ contains the first k eigenvectors of the correlation matrix $\boldsymbol{C}_{uu}$ corresponding to the eigenvalues $\lambda_1, .., \lambda_k$ and n is the number of discretization points of the random field, i.e. the dimension of the correlation matrix. As a consequence, the transformation

$$\boldsymbol{u} = \sum_{i=1}^{k} \boldsymbol{\Phi}_i z_i, \tag{5.38}$$

allows to simulate uncorrelated Gaussian variables z_i and transform them back to the correlated Gaussian space using Eq. (5.38). Finally, these variables are transformed to their original distribution similar to the procedure in section 5.2.3.1. The advantage of this spectral decomposition is that it is often sufficient to calculate only the largest k eigenvalues of the covariance matrix C_{uu} to accurately approximate the full covariance matrix. The number of required eigenvalues depends on the correlation length and the discretization. In general, it can be stated that the smaller the ratio between correlation length and mesh size, the more eigenvectors have to be included.

A simplified approach is the midpoint method [174], where the random field within a single element is assumed to be constant and represented by a random variable at the center of the element.

Another approach is based on a local averaging procedure following [185], where the random field in a certain region Ω_i (e.g. an element) is represented by its average value

$$U_i = \frac{1}{\Omega_i} \int_{\Omega_i} U(\boldsymbol{x})\, d\Omega. \tag{5.39}$$

For homogeneous random fields with rectangular non-overlapping domains Ω_i, [186] developed formulas for the autocorrelation function between the random variables U_i. One problem of the method is that the pdf of the random variables U_i can only be derived for gaussian random fields [187].

A discretization of a random field can also be performed with the shape function method [188], where the discretization of a random field is done on a separate grid and a mapping to the finite element mesh is performed. In combination with finite elements, the random field is often discretized at the nodes and the FE-shape functions are used for the interpolation.

A final class of methods are series expansion methods, where the random field is represented by a series expansion of functions with random coefficients, e.g. the Karhunen-Loéve transform.

5.3.4 Discretization using a series expansion approach

In the current implementation, an approach developed by [189, 190] has been applied, which allows the fast sampling of a random field using the Fast Fourier Transform.

Starting from the cross-correlation function, the corresponding cross-spectral density matrix is obtained by using the n-dimensional Whiener-Khintchine relations [191]

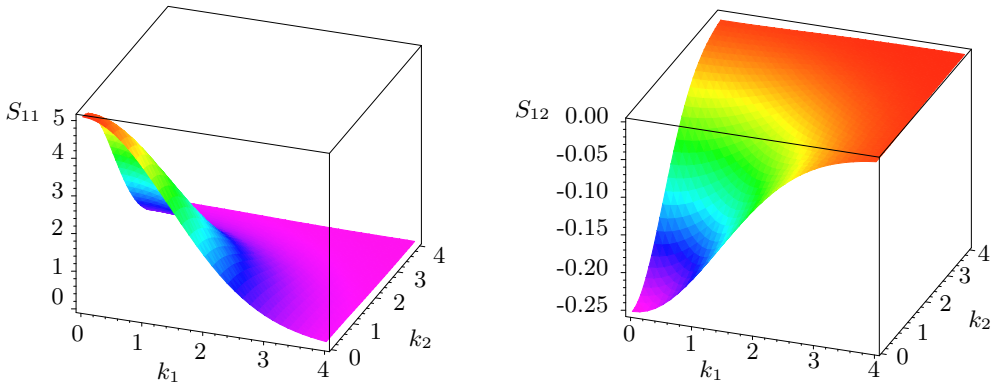
$$S_{jk}(\boldsymbol{k}) = \frac{1}{(2\pi)^n} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} R_{jk}(\boldsymbol{\xi}) \boldsymbol{e}^{-i\boldsymbol{k}\cdot\boldsymbol{\xi}} d\boldsymbol{\xi}, \tag{5.40}$$

with $\boldsymbol{k}\cdot\boldsymbol{\xi}$ being the dot-product between the n-dimensional wave number vector $\boldsymbol{k}$ and the space lag vector $\boldsymbol{\xi}$. From the auto-correlation function in Eq. (5.33), the cross-correlation function is calculated for a 2D random field (n=2):

$$S_{jk}(\boldsymbol{k}) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \rho_{jk}\sigma_j\sigma_k \boldsymbol{e}^{-\left(\frac{\xi_1^2+\xi_2^2}{l_h^2}\right)} \boldsymbol{e}^{-i\boldsymbol{k}\cdot\boldsymbol{\xi}} d\xi_1 d\xi_2. \tag{5.41}$$

Using the formula of Euler $\boldsymbol{e}^{x+iy} = \boldsymbol{e}^x\left[\cos(y) + i\sin(y)\right]$, the symmetry of the cosine and the antisymmetry of the sine function and the relation $\int_0^\infty \boldsymbol{e}^{-\alpha^2x^2}\cos(\beta x) = \frac{\sqrt{\pi}}{2\alpha}\boldsymbol{e}^{-\frac{\beta^2}{4\alpha^2}}$, the following expression is obtained:

$$S_{jk}(\boldsymbol{k}) = \frac{\rho_{jk}\sigma_j\sigma_k l_h^2}{4\pi}\, \boldsymbol{e}^{-\frac{l_h^2}{4}\left(k_1^2+k_2^2\right)}. \tag{5.42}$$



(a) power spectral density S_{11} (b) cross spectral density S_{12}

Figure 5.7 Spectral density function for two variables with $\sigma_1 = 10$, $\sigma_2 = 1$, $l = 1$ and $\rho_{12} = -0.5$.

Note that the integration of ξ_1 and ξ_2 can be performed independently, which is the reason for choosing the special form of the cross-correlation function in Eq. (5.33). An exemplary plot of the spectral density function is given in Fig. 5.7.

The cross-spectral density matrix can be decomposed using a Cholesky decomposition

$$\boldsymbol{S}(\boldsymbol{k}) = \boldsymbol{H}(\boldsymbol{k})\boldsymbol{H}^T(\boldsymbol{k}) \tag{5.43}$$

where $\boldsymbol{H}(\boldsymbol{k})$ is a lower triangular matrix.

The r^{th} component of a 2D-mV Gaussian field $\boldsymbol{f}(\boldsymbol{x})$ at location $\boldsymbol{x}$ with zero mean can be simulated by

$$f_r = 2\sum_{s=1}^{m}\sum_{l_1=1}^{N_1}\sum_{l_2=1}^{N_2}\sum_{\substack{I_1=1\\I_2=\pm 1}} H_{rs}(k_1^{s,l_1}, k_2^{s,l_2})\sqrt{\Delta k_1 \Delta k_2}\cos\left(I_1 k_1 x_1 + I_2 k_2 x_2 + \Psi_{s,I_2}^{l_1,l_2}\right), \tag{5.44}$$

where $\Delta k_i = \dfrac{k_{iu}}{N_i}$ with k_{iu} being the upper cut-off frequency beyond which the cross-spectral density matrix is assumed to be zero and N_i being the number of subdivisions in direction i. $\Psi_{s,I_2}^{l_1,l_2}$ are $m \cdot 2^{(n-1)}$ sets of $N_1 N_2 \cdots N_n$ independent random phase angles uniformly distributed in the interval $[0, 2\pi]$. The scalars $H_{rs}(k_1^{s,l_1}, k_2^{s,l_2})$ are elements from Eq. (5.43) evaluated at

$$k_i^{s,l_i} = (l_i - 1)\Delta k_i + \frac{s}{m}\Delta k_i. \tag{5.45}$$

For each s, there is a frequency shift of $\frac{\Delta k_i}{m}$, which is required for the process to be ergodic [189]. The scalar fields f_r are periodic with the period given by

$$L_{0,i} = m\frac{2\pi}{\Delta k_i} \tag{5.46}$$

and asymptotically Gaussian due to the central limit theorem for $N_i \to \infty$ [192]. It can further be proven that every sample function is ergodic in the mean and the correlation as long as the spatial discretization is a multiple of the period $L_{0,i}$.

5.3.4.1 Use of the Fast Fourier Transform

The application of the Fast Fourier Transform allows to increase the efficiency of the numerical implementation. For that purpose, the function f_r is not sampled at random positions $\boldsymbol{x}$, but on a regular grid with length $L_{0,i}$ and spacing Δx_i. The spacing Δx_i is calculated from

$$M_i \Delta x_i = \frac{2\pi}{\Delta k_i} \tag{5.47}$$

with the condition $M_i \geq 2N_i$ to avoid aliasing [193].

Equation (5.44) can be rewritten as

$$f_r(p_1\Delta x_1, p_2\Delta x_2) = \sum_{s=1}^{r} \mathrm{Re}\left\{ g_{rs}\left(q_1\Delta x_1, q_2\Delta x_2\right) \exp\left(\sum_{k=1}^{n} i p_k \Delta x_k \frac{s\Delta k_k}{m} \right) \right\}. \tag{5.48}$$

The auxiliary functions $g_{rs}\left(q_1\Delta x_1, q_2\Delta x_2\right)$ are evaluated in a reduced domain with

$$q_k = p_k \bmod M_k, \tag{5.49}$$

i.e. q_k is the remainder when dividing p_k by M_k and

$$g_{rs}\left(q_1\Delta x_1, q_2\Delta x_2\right) = \sum_{l_1=0}^{M_1-1} \sum_{l_2=0}^{M_2-1} \sum_{I_2 \pm 1} B_{r,s,I_2}^{l_1,l_2} \exp\left(\sum_{k=1}^{n} i q_k \Delta x_k l_k \Delta k_k I_1 \right) \tag{5.50}$$

$$B_{r,s,I_2}^{l_1,l_2} = 2H_{rs}(k_1^{s,l_1}, k_2^{s,l_2}) \sqrt{\Delta k_1 \Delta k_2} \exp\left(i\Psi_{s,I_2}^{l_1,l_2} \right). \tag{5.51}$$

The elements H_{rs} are calculated at the same locations as in Eq. (5.44), but in order to account for the fact that the summation starts at zero, the wave numbers are calculated by

$$k_i^{s,l_i} = l_i \Delta k_i + \frac{s}{m} \Delta k_i. \tag{5.52}$$

When rearranging Eq. (5.50) using Eq. (5.47), the summation can be efficiently performed by using the Fast Fourier Transform (FFT):

$$g_{rs}\left(q_1\Delta x_1, q_2\Delta x_2\right) = \underbrace{\sum_{I_2=\pm 1} \sum_{l_1=0}^{M_1-1} \underbrace{\sum_{l_2=0}^{M_2-1} B_{r,s,I_2}^{l_1,l_2} \exp\left(\frac{2\pi i q_2 l_2}{M_2} I_1 \right)}_{\text{1-dimensional FFT}} \exp\left(\frac{2\pi i q_1 l_1}{M_1} I_1 \right)}_{\text{2-dimensional FFT}}. \tag{5.53}$$

As illustrated in Eq. (5.53), the 2-dimensional FFT is calculated using the row-column method, which consists of solving a sequence of 1-dimensional FFT's. In the implementation, the software package [194] was used to calculate the 1-dimensional FFT's.

Determination of upper cut-off wave number

In order to determine the upper cut-off wave number to be included, a procedure based on [193] has been used. A parameter $\epsilon \ll 1$ is defined such that by cutting the wave number after k_u, the corresponding volume under the power spectral density function in Fig. 5.7 is only a fraction $1-\epsilon$ of the total volume.

$$\int_0^{k_u}\int_0^{\infty} S(k_1,k_2)\, dk_1 dk_2 = (1-\epsilon)\int_0^{\infty}\int_0^{\infty} S(k_1,k_2)\, dk_1 dk_2 \tag{5.54}$$

Using the cross correlation function given in Eq. (5.41), the following equation is obtained:

$$\frac{\sigma^2}{4}\mathrm{erf}\left(\frac{l_h k_u}{2}\right) = (1-\epsilon)\frac{\sigma^2}{4}, \tag{5.55}$$

where $\mathrm{erf}(k_u)$ is the error function

$$\mathrm{erf}(k_u) = \frac{2}{\sqrt{\pi}}\int_0^{k_u} e^{-k^2} dk. \tag{5.56}$$

In order to determine the cut-off wave number, an asymptotic expansion is used in combination with a bisection algorithm

$$\mathrm{erf}(k_u) \approx 1 - \frac{e^{-k_u^2}}{k_u\sqrt{\pi}}\left(1+\sum_{n=1}^{\infty}(-1)^n\frac{(2n)!}{n!(2k_u)^n}\right). \tag{5.57}$$

In the numerical implementation, a value of $\epsilon = 10^{-4}$ has been used.

5.3.4.2 Numerical implementation

The power spectral density matrix from Eq. (5.42) can be decomposed into

$$S_{jk}(\boldsymbol{k}) = \frac{\rho_{jk}\sigma_j\sigma_k l_h^2}{4\pi}\, \boldsymbol{e}^{-\frac{l_h^2}{4}\left(k_1^2+k_2^2\right)} = \bar{\boldsymbol{S}}^0_{jk} s(\boldsymbol{k}) \tag{5.58}$$

with

$$\bar{\boldsymbol{S}}^0_{jk} = \frac{\rho_{jk}\sigma_j\sigma_k l_h^2}{4\pi} \tag{5.59}$$

$$s(\boldsymbol{k}) = \boldsymbol{e}^{-\frac{l_h^2}{4}\left(k_1^2+k_2^2\right)}. \tag{5.60}$$

It is to be noted that $s(\boldsymbol{k})$ is only a scalar value. Consequently, the calculation of the Cholesky-decomposition of $S_{jk}(\boldsymbol{k})$ for different $\boldsymbol{k}$ can be simplified to

$$S_{jk}(\boldsymbol{k}) = \sqrt{s(\boldsymbol{k})}^2 \boldsymbol{H}^0\boldsymbol{H}^{0T} \tag{5.61}$$

$$S^0_{jk} = \boldsymbol{H}^0\boldsymbol{H}^{0T}. \tag{5.62}$$

The Cholesky-decomposition of S^0_{jk} is calculated once in advance, and afterwards, multiplied by $\sqrt{s(\boldsymbol{k})}$ for each $\boldsymbol{k}$.

The number of wave numbers N_i is determined by two conditions. First of all, N_i should be large enough to accurately model the correct standard deviation of the random field (in our application 256 has been used as minimum). A second condition is that the random field is periodic with

$$L_{0,i} = \frac{m 2\pi N_i}{k_u} > L_i \tag{5.63}$$

Since the length of the period $L_{0,i}$ should be larger than the actual sample length L_i, a second condition for N_i is given, although in the present examples this condition was always fulfilled.

The described algorithm yields a discretization of the random field at points on a regular grid with spacing Δx_i. However, the integration points of the elements are, in general, located irregularly within the grid. Consequently, an interpolation between the grid points is performed. For this purpose, a Moving Least Squares (MLS) interpolation with linear basis functions, an exponential weighting function and an influence radius corresponding to the diagonal in the grid $\sqrt{\Delta x_1^2 + \Delta x_2^2}$ has been used. This small influence radius reduces the interpolation to an almost linear interpolation, but avoiding the discontinuity in the first derivative of a bilinear interpolation on the regular grid. Using this grid interpolation results in an additional constraint, namely, that the spacing between grid points is smaller than a certain length, which is correlated to the correlation length. In the implementation, a constraint $\Delta x_i < l_h/3$ using Eq. (5.47) has been used.

In Algorithm 5.1, the procedure for creating samples of a stochastic field using m Gaussian variables in 2 dimensions is summarized.

5.3.4.3 Example

In the present example, a 2-dimensional (n=2) random field with two correlated variables (m=2) is simulated. The FE-model is a rectangular area (20cm x 20cm) with an element length of 1cm. The two random fields F_1 and F_2 have a standard deviation of $\sigma_1 = 0.1$ and $\sigma_2 = 5$ and are correlated with a correlation coefficient of 0.8 and a correlation length of 5cm. The random field was sampled 1000 times. Crosscorrelation and autocorrelation for all pairs of integration points were calculated. Due to the fact that for each distance several values are obtained (there are many integration points having the same distance to each other), and only discrete values are actually sampled, a smoothing using an MLS interpolation with an influence radius of 1cm has been performed. In order to reduce the number of integration points, only one point per element has been used. This restriction only reduced the number of pairs to check the method, the speed of the sampling was only slightly decreased (smaller number of interpolation points). The results are illustrated in Fig. 5.8. Obviously, the sampling correlations match the target correlations. The sampling of a single sample took about 3 seconds on a standard PC (Pentium 4, 3.40GHz, 2GB RAM). The size of the specimen is not important, since the sampling grid (in this case the grid was 1461.63cm x 1461.63cm) is much larger than the specimen dimension. Compared to the

Algorithm 5.1 Sampling of a multivariate multidimensional stochastic field

1. Determine cut-off wave number by iteration and bisection until

$$\frac{\sigma^2}{4}\mathrm{erf}\left(\frac{l_h k_u}{2}\right) < (1-\epsilon)\frac{\sigma^2}{4}$$

2. Calculate Cholesky decomposition of $S^0_{jk} = \boldsymbol{H}^0 \boldsymbol{H}^{0T}$
3. Calculate number of wave numbers to be considered

$$N_i = \max\left(\frac{k_u L_i}{m 2\pi}, 256\right)$$

4. Determine wave number increment $\Delta k_i = \dfrac{k_u}{N_i}$
5. Determine number of sample points M_i in spatial direction

$$M_i = \max\left(\frac{3 \cdot 2\pi}{l_h \Delta k_i}, 2N_i\right)$$

6. Compute auxiliary functions $g_{rs}\left(q_1\Delta x_1, q_2 \Delta x_2\right)$ using 2-dimensional FFT

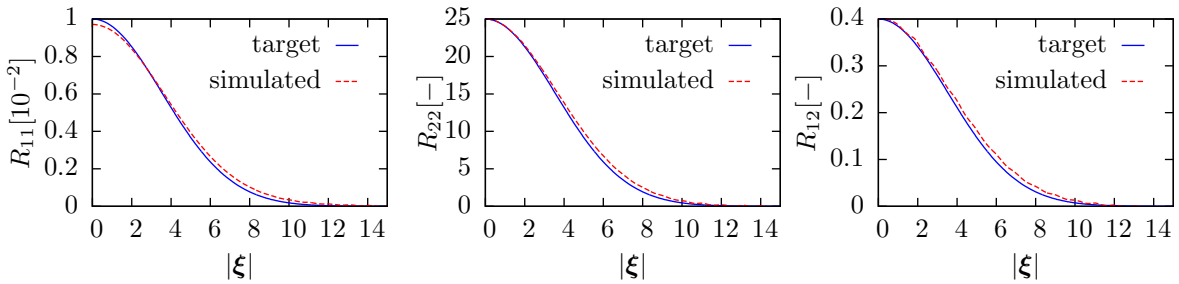
$$g_{rs}\left(q_1\Delta x_1, q_2 \Delta x_2\right) = \sum_{I_2 \pm 1} \sum_{l_1=0}^{M_1-1} \sum_{l_2=0}^{M_2-1} B^{l_1,l_2}_{r,s,I_2} \exp\left(\frac{2\pi i q_2 l_2}{M_2} I_1\right) \exp\left(\frac{2\pi i q_1 l_1}{M_1} I_1\right)$$

$$B^{l_1,l_2}_{r,s,I_2} = 2H^0_{rs}\sqrt{s(k_1^{s,l_1}, k_2^{s,l_2})}\sqrt{\Delta k_1 \Delta k_2}\exp\left(i\Psi^{l_1,l_2}_{s,I_2}\right).$$

7. Calculate random field at discrete grid points ($q_i = p_i \bmod M_i$)

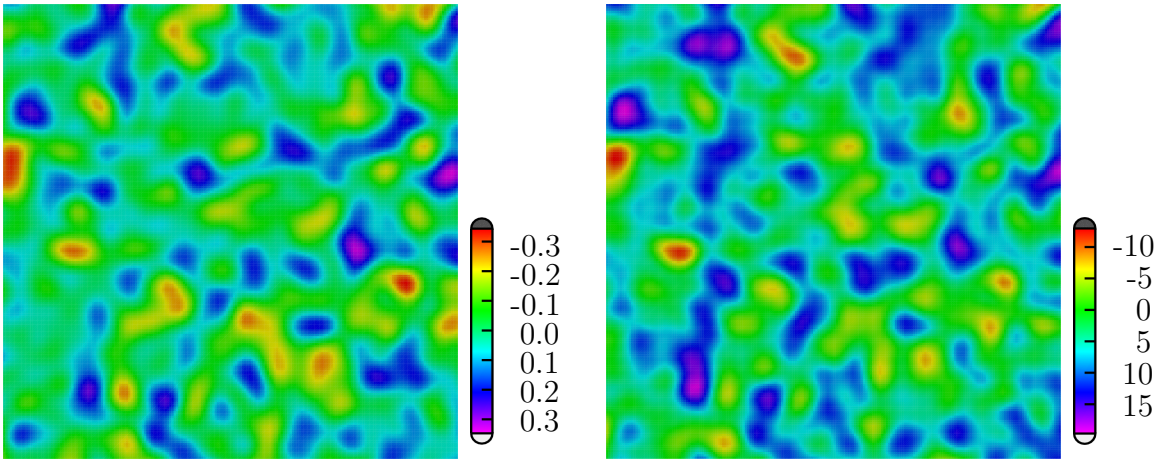
$$f_{Gr}(p_1\Delta x_1, p_2 \Delta x_2) = \sum_{s=1}^{r} \mathrm{Re}\left\{ g_{rs}\left(q_1\Delta x_1, q_2 \Delta x_2\right) \exp\left(\sum_{k=1}^{n} i p_k \Delta x_k \frac{s\Delta k_k}{m}\right)\right\}.$$

8. Interpolate random field for integration points using MLS on the regular grid



(a) autocorrelation F_1 (b) autocorrelation F_2 (c) crosscorrelation F_1, F_2

Figure 5.8 Comparison of auto- and crosscorrelation function for 1000 realization of the random field with the target curves.



(a) F_1 (b) F_2

Figure 5.9 Sample of a 2-D random field with two variables, correlation 0.8, $\sigma_i = [0.1, 5]$, correlation length 5cm on a domain 100cm x 100cm.

eigenvalue decomposition of the covariance matrix, this approach is faster for a large number of integration points. For illustration purposes, a sample of the discretized random field is shown in Fig. 5.9 with an increased mesh size of 100cm x 100cm. The correlation between the variables is clearly observable. The presented procedure simulates random fields with zero mean. A random field with nonzero mean can be simulated by simply adding the mean value to all the samples.

5.4 Example - tensile test

The stochastic variability of material or geometrical parameters in the numerical model is represented using stochastic finite elements [195, 196]. In general, the material parameters of the constitutive equation or parameters describing the geometry, e.g. the thickness of a 2D element, are simulated using random fields as described in the previous section. Afterwards, a Monte Carlo simulation of the random field is performed with different sets of material parameters. Although this requires many computations (which can be effectively parallelized), this allows a reliable investigation of the influence of stochastic parameters on the statistical character of the response.

Engineers who perform numerical simulations often try to adapt their model until a fit between simulated and experimental data is obtained. Due to cost of experiments, often only a few experiments are performed, which does not allow an appropriate characterization of the stochastic response. As will be demonstrated in the following example, the stochastic scatter of the response for mesoscale models has to be taken into account, e.g. by the procedure described in the previous section. Several examples of macroscopic simulation of concrete using random fields can be found in the literature [197, 198, 199, 200, 201]. However, an

experimental verification of stochastic parameters such as correlation coefficients, correlation length or coefficient of variation in the numerical model is seldom given.

For mesoscale models of concrete, experimental data describing the spatial variability of material parameters such as Young's modulus, fracture energy or tensile strength are not available. This can, on the one hand, certainly be attributed to problems related to the technical realization, and, on the other hand, to the fact that the application of mesoscale models to concrete is relatively new.

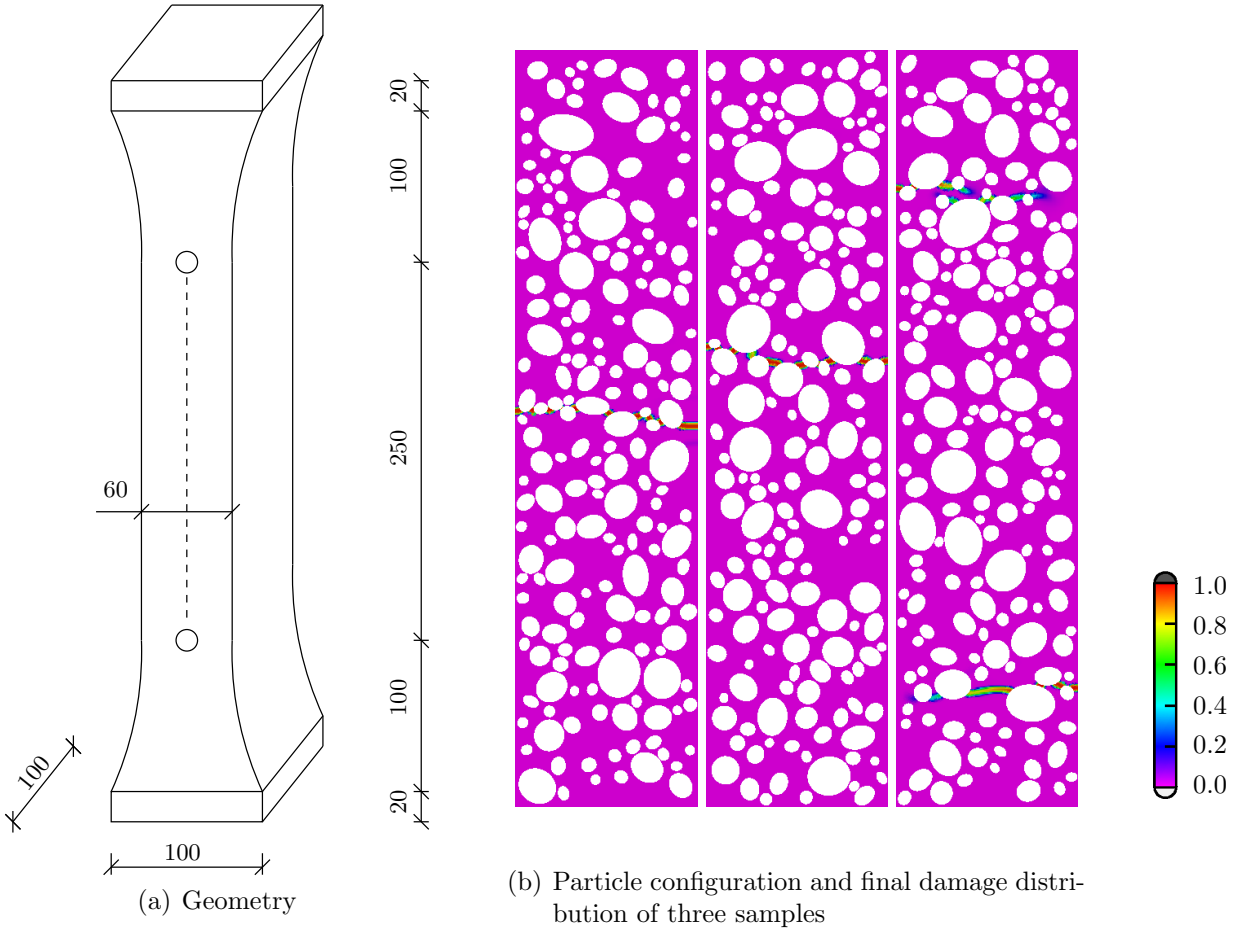
Due to the absence of experimental data describing the spatial correlation between material points on the mesoscale, sensitivity studies are performed in order to show the influence of certain parameters and the general applicability of the method.

The correlation length is one of the key parameters in the stochastic description of the material and its origin can certainly be attributed to several phenomena. On the one hand, one might argue that the correlation length is related to the smallest aggregate size explicitly modeled in the simulation (or similarly the largest aggregate size homogenized in the matrix). In this case, the origin of the correlation is due to the homogenization procedure of the matrix. On the other hand, a correlation length related to structural dimensions seems also to be reasonable. Here, the spatial correlation between material points is e.g. due to the manufacturing process. Theoretically, both effects can be simulated by a superposition of two random fields each having a separate standard deviation and correlation length.

As an example, a tensile test performed by [202] was used. The geometry is illustrated in Fig. 5.10. The main advantage of this test compared to many others is the fact that there is no prescribed notch, which triggers the localization. As a consequence, the particle distribution and the stochastic distribution of material parameters has a much more pronounced but nonetheless more realistic effect compared to a notched specimen. In order to simplify the numerical model, only the inner part with a width of 60cm is modeled, which corresponds to the region over which the relative displacements are measured in the experiment. Strong boundary effects, e.g. that damage develops especially in the boundary layer triggered by the stiffer displacement constraints (compared to the experiment) was not observed. The material parameters used in the simulation are summarized in Table 5.3. The grading curve was taken from [202], where the total amount of aggregates is given by 77 mass-%, partitioned into 24.6 mass-% in the class 8/16, 29.2 mass-% in the class 2/8 and 23.1 mass-% homogenized in the matrix material with a diameter smaller than 2mm.

5.4.1 Influence of particle distribution

In a first test, the influence of the particle distribution is investigated. On the basis of 50 simulations with different aggregate distributions, the influence on the load-displacement curve is evaluated. In Fig. 5.10(b) the aggregate distribution and the damage at the final load step for the first three samples is illustrated. It is observed that the particle configuration has a substantial influence on the location of the final crack. Furthermore, cracks are passing primarily along grain boundaries, and matrix cracks are only obtained when these cracks in the interface layer coalesce. In the initial stage, many microcracks depicted by zones of local damage are obtained. In general, the final failure mode is a single crack which has



(a) Geometry

(b) Particle configuration and final damage distribution of three samples

Figure 5.10 Geometry and final damage distribution of the tensile test performed by [202] for varying aggregate distributions.

been developed from these microcracks. It sometimes happens - as in the right sample in Fig. 5.10(b) - that two competing cracks are obtained and the localization into a single crack is obtained only at an advanced loading level.

The mean and the standard deviation of the corresponding load-displacement curves are illustrated in Fig. 5.11. It is observed that the particle distribution primarily influences the post-peak branch - the peak load is only slightly influenced. This is due to the fact that the particles primarily influence the crack path and, consequently, the length over which energy is dissipated. Using varying specimen dimensions, [151] showed that only from the particle distribution size effects can be explained.

5.4.2 Influence of correlation length

In a second test, the additional effect of spatially variable material parameters is investigated. Due to the lack of experimental data, a sensitivity study is performed, which demonstrates the influence of certain parameters and illustrates the overall applicability of the method. The

Table 5.3 Material parameters for the numerical simulation of the tensile test [202].

parameter			matrix	interface	aggregate
Young's modulus	E	$\left[\frac{N}{mm^2}\right]$	$\approx 27500^a$	-	54000
Poisson's ratio	ν	$[-]$	0.18	-	0.2
tensile strength	f_{ct}	$\left[\frac{N}{mm^2}\right]$	3.9	2.93	-
uniaxial compressive strength	f_c	$\left[\frac{N}{mm^2}\right]$	39	-	-
biaxial compressive strength	f_{c2}	$\left[\frac{N}{mm^2}\right]$	45.2	-	-
fracture energy	G	$\left[\frac{Nmm}{mm^2}\right]$	0.219	0.164	-
nonlocal radius	r	$[mm]$	1.5	-	-
interaction value	α	$[-]$	-	1	-

[a]The parameter was calculated from the macroscopic Young's modulus 33930 $\frac{N}{mm^2}$ and the actual fraction of aggregates using Eq. (4.20).

material parameters Young's modulus, tensile strength and uniaxial compressive strengths are modeled by a random field with a correlation coefficient of 0.5 and a variation coefficient of 0.15. The correlation length was varied in the range of 8mm and 1024mm. The influence of the correlation length on the macroscopic tensile strength is illustrated in Fig. 5.12. In the limit with an infinite correlation length, a constant material parameter within the full specimen is obtained. This is plotted at l_{corr}=2048mm, where the material parameters have been simulated by correlated random variables (no random field). The other limit is the case of an almost vanishing correlation length, implying that each material point is independent of the neighboring point. From a numerical implementation point of view, this requires that the mesh size is smaller than the correlation length in order not to homogenize the material parameters within an element, since the complexity of the approximation is limited by the applied ansatz of the shape functions. For a constant mesh size and a decreasing correlation length, the limit is probably related to a constant distribution of parameters. The element can be interpreted as an RVE and, in the limit of a vanishing correlation length, the response of the RVE is homogeneous, since the numerical integration performed over the (infinite number of) integration points homogenizes all fluctuations of the random field. This limit is plotted at l_{corr}=2mm.

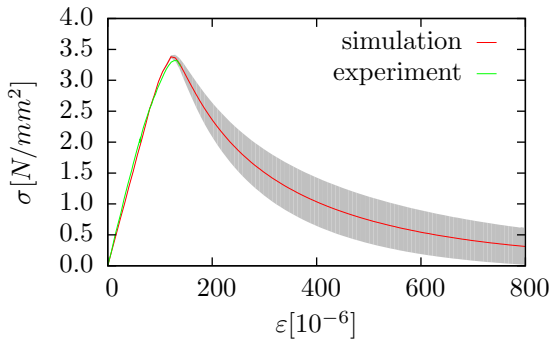
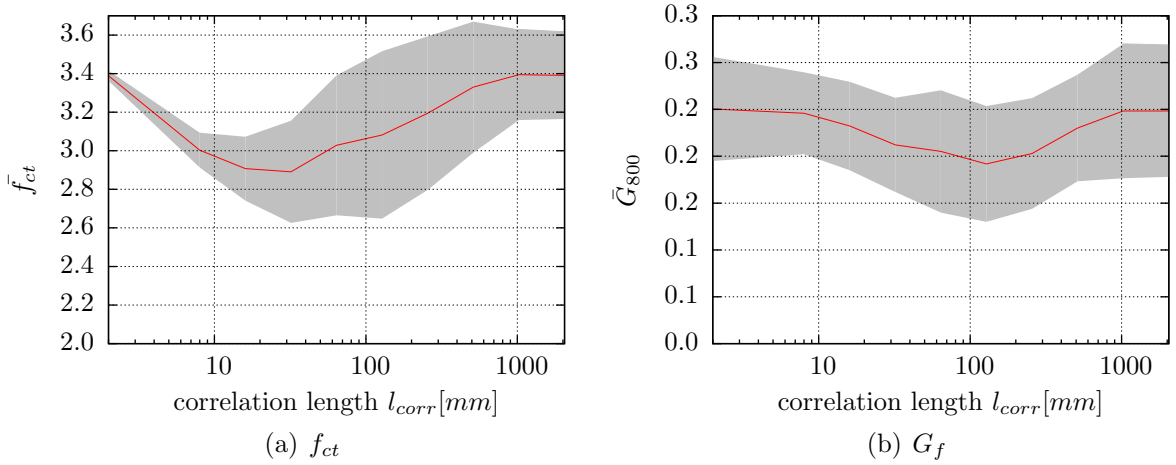


Figure 5.11 Influence of particle distribution on the load-displacement curve.



(a) f_{ct} (b) G_f

Figure 5.12 Influence of correlation length on the global tensile strength and the fracture energy.

It is observed in Fig. 5.12(a) that the mean of the tensile strength is almost identical at the limits, where the material parameters are not varied and the limit, where a correlated set of random variables is used. The major difference is characterized by the standard deviation. It is almost zero for the case where no random field is used. This is probably due to the fact that the size of the specimen is sufficiently large. Consequently, the particle distribution has no influence on the global peak load. This implies that the peak load is mainly influenced by the tensile strength of the matrix and the particles. At the right limit with a constant parameter distribution in the specimen, the variation coefficient of 0.05 is much smaller than the prescribed correlation coefficient of 0.15. This is due to the constant tensile strength of the interface, which has a significant influence. It is furthermore observed that a minimum of the tensile strength is observed at a correlation length of approximately 32mm, where the mean of the tensile strength is reduced by 15%. The correlation length corresponding to the minimum tensile strength is related to the width of the specimen, since a correlation length in the order of the width of the specimen weakens a full cross section. The reduction of the mean tensile strength is primarily influenced by the length of the specimen (the longer the specimen, the higher the probability of having a weak section) and the variation coefficient of the material parameters. For the fracture energy, the general trend of the mean is similar, at both limits identical values and a reduced mean of 20% at l_{corr}=128mm. The reduction at l_{corr}=128mm can be explained by the influence of the distributions of Young's modulus and tensile strength on the crack path, which is less influenced by the particle distribution and, consequently, is less curved. As a result, the fracture surface decreases and hence the global fracture energy. It is observed that the standard deviation is almost constant and independent of the correlation length. This is due to the fact that the fracture energy was assumed to be constant (in this sensitivity analysis) and the main influence on the standard deviation of the fracture energy has the particle distribution, which is independent of the correlation length.

Summarizing the preceding example, it can be concluded that spatially variable material parameters have a significant influence on the mean and the stochastic scatter of the numerical

result, and can be used to explain the stochastic scatter in experimental tests. However, a principal problem related to the experimental determination of these parameters remains, e.g. the correlation length, variation coefficients and correlation coefficients have to be determined in order to apply them in a reliable numerical simulation.

Chapter 6

Metamodels

6.1 Motivation

Growing computer capacities increase the potential of numerical simulations. However, due to the increased level of accuracy in these models, the overall simulation time remains almost constant. For certain applications as e.g. investigations concerning robustness or reliability, or the coupled computation on different scales, it can be advantageous to perform several numerical simulations at once covering a large range of possible input parameters, and then replace the actual complicated model by a metamodel, which is much faster to evaluate. This offers the possibility to use complex models wile keeping the overall computation time low. On the one hand, such an approach requires the metamodel to generalize well, which means that it should give a reasonable response for points that do not coincide with training samples. On the other hand, the training samples should be reproduced as best as possible. Both of these conditions are contradictory, since the best metamodel for an accurate reproduction of the training samples is very complex with many free parameters, while, on the other hand, a metamodel with only a few parameters generalizes best. In order to find the balance between these two conditions, a parameter fitting of the metamodel is often required, which might be e.g. the influence radius, the number and position of basis functions, the number of neurons and hidden layers or certain precision values.

In this chapter, the metamodels used in this work are presented. Furthermore, strategies for the parameter fitting of these models are discussed.

6.2 Neural networks

6.2.1 Introduction

Artificial neural networks are inspired by their biological counterpart in the brain, where as many as 10^{10} neurons are connected with each other and build up an efficient system for the processing of information. In a similar way, artificial neural networks (in the following simply labeled neural networks) are composed of a set of neurons, which exchange information. Mathematically, this system can be interpreted as a graph, with the neurons being the nodes. From the total set of neurons, a set of input neurons and a set of output neurons can be defined, which enter information into the system and, vice versa, extract information from the system.

The original ideas to mathematically represent the biological structure of the brain date back to McCulloch [203]. In the following, Hebb introduced a training algorithm for the networks known as Hebb learning [204]. Another pioneering work was done by Rosenblatt, which invented the perceptron and an appropriate learning rule [205]. Considerable decrease in the activity of research in this area followed the publication of Minsky and Papert [206], where it was proven that there is no two-layer perceptron that can solve the XOR problem. Only after the introduction of the backpropagation algorithm as an efficient tool for supervised learning of multilayer perceptrons, the research activity restarted and neural networks were applied in many different areas such as e.g. pattern recognition, function approximation, signal filtering and data compression.

Nowadays, a variety of different networks types exist, e.g. recurrent networks such as Hopfield networks [207] or Kosko networks [208], networks with higher order neurons such as Sigma-Pi neurons [209] or self organizing maps [210] as an example for unsupervised learning. In general, these networks can be distinguished with respect to their structure, the dynamics in the execution mode and the learning methods.

In the following, only the multilayer perceptron as a special type of neural networks for metamodeling is considered. In this context, Hornik demonstrated that a multilayer perceptron with a single hidden layer is a universal approximator [211], i.e. that any continuous function can be approximated on a compact support to any user-defined accuracy by a multilayer perceptron with a single hidden layer.

6.2.2 Architecture of a multilayer perceptron

The multilayer perceptron as one kind of neural networks consists of an input layer, an output layer and a certain number of hidden layers. The number of neurons in the input layer corresponds to the dimension of the input vector, whereas the dimension of the output layer corresponds to the dimension of the output vector. In this work, a fully connected network is used, i.e. each neuron is connected to all the neurons of the previous layer. The output of a neuron i in layer j is then given by

$$o_i^{(j)} = f\left(\sum_{k=1}^{N^{(j-1)}} w_{ki}^{(j)} o_k^{(j-1)} + b_i^{(j)}\right), \tag{6.1}$$

where $N^{(j)}$ is the number of neurons in layer j and f is a transfer function. For the input layer with $j = 0$, the output is equal to the input parameter corresponding to that neuron. Common choices for the transfer function in the hidden layer are e.g. sigmoidal functions

$$f^s(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \tag{6.2}$$

whereas for the output layer linear functions are used. The network architecture with two inputs, two outputs and a hidden layer with three neurons is illustrated in Fig. 6.1.

The free parameters of the network, which are the weights $w_{ij}^{(k)}$ and the biases $b_i^{(k)}$ are calculated from given associations $\mathcal{D} = \left\{\boldsymbol{x}^{(i)}, \boldsymbol{t}^{(i)}\right\}$, which are called training samples. In the

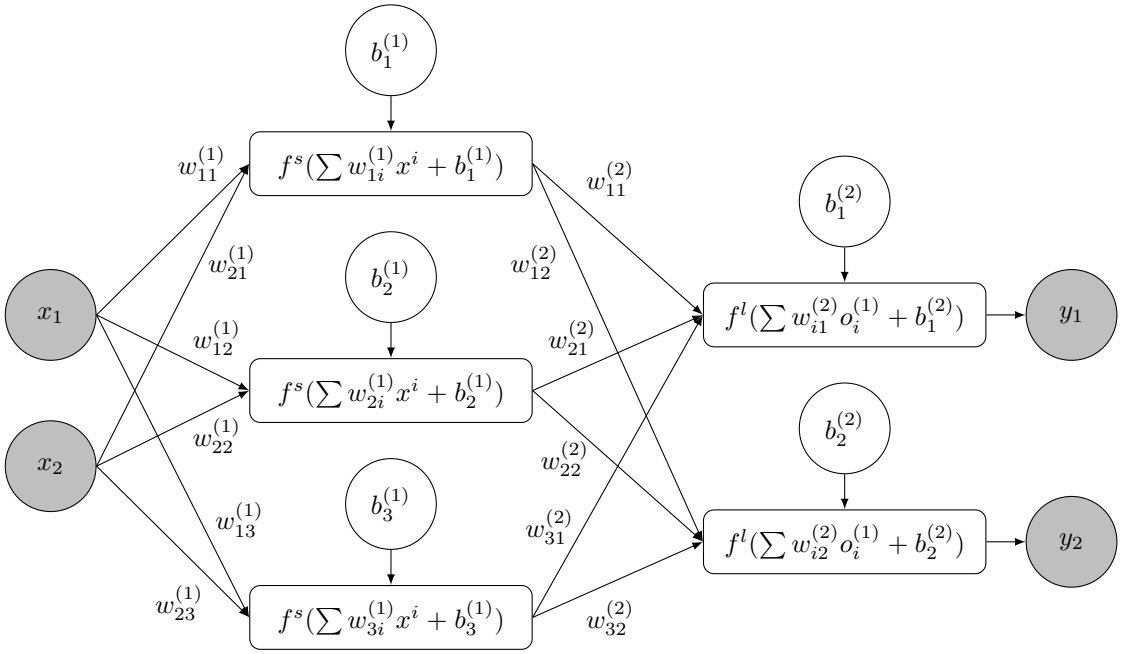


Figure 6.1 Architecture of a multilayer perceptron.

following, these free parameters are placed in a vector $\boldsymbol{w}$ with components w_i. For each of these training samples, the error $\boldsymbol{e}^{(m)}(\boldsymbol{w})$ between the approximation of the neural network and the exact value can be calculated. The squared sum over the errors of all training samples can be used to identify the quality of the approximation and the weights and biases are calculated to minimize this sum of squared errors.

$$\min F(\boldsymbol{w}) = \sum_{i=1}^{M} \frac{1}{2} \left| \boldsymbol{e}^{(m)} \right|^2 \tag{6.3}$$

$$\boldsymbol{e}^{(m)}(\boldsymbol{w}) = \boldsymbol{t}^{(m)} - \boldsymbol{y}^{(m)}(\boldsymbol{w}, \boldsymbol{x}^{(m)}) \tag{6.4}$$

where M is the number of training samples and $\boldsymbol{y}^{(m)}(\boldsymbol{w}, \boldsymbol{x}^{(m)})$ is the output of the neurons in the output layer. This approach is called batch mode, since the sum of errors for all training samples is minimized. A different approach, not followed in this thesis, is the sequential or online approach, where the error is calculated for a single sample and the weights are adapted accordingly. Repeating this for all samples, often with a random order, gives one cycle. After a certain number of cycles or until another convergence criteria is fulfilled, the algorithm stops. However, this approach has the disadvantage that convergence is not assured and that after a cycle the objective function might increase.

Different algorithms can be used to determine the free parameters such as the standard backpropagation algorithm [212], RPROP [213], the conjugate gradient method [214] and the scaled conjugate gradient algorithm [215]. In this paper, the Levenberg-Marquardt algorithm [216] and a preconditioned conjugate gradient approach has been used.

6.2.3 Training algorithms

6.2.3.1 Levenberg-Marquardt algorithm

The Levenberg-Marquardt algorithm is a modified Newton method. It requires the calculation of the gradient and the hessian of the error functions with respect to the weights. The gradient is given by

$$\boldsymbol{G} = \frac{\partial F}{\partial w_j} = \sum_{m=1}^{M} \sum_{i=1}^{N^o} e_i^{(m)} \frac{\partial e_i^{(m)}}{\partial w_j} = -\sum_{q=1}^{M} \boldsymbol{J}^{(m)^T} \boldsymbol{e}^{(m)} \tag{6.5}$$

$$\boldsymbol{J}^{(m)} = \begin{bmatrix} \frac{\partial y_1^{(m)}(\boldsymbol{w})}{\partial w_1} & \cdots & \frac{\partial y_1^{(m)}(\boldsymbol{w})}{\partial w_n} \\ \vdots & & \vdots \\ \frac{\partial y_N^{(m)}(\boldsymbol{w})}{\partial w_1} & \cdots & \frac{\partial y_N^{(m)}(\boldsymbol{w})}{\partial w_n} \end{bmatrix}, \tag{6.6}$$

where $\boldsymbol{J}$ describes the sensitivity of the outputs with respect to the free parameters calculated with the backpropagation algorithm and N^o is the dimension of the output vector. The Hessian can be expressed in a similar way as

$$\boldsymbol{H} = \frac{\partial^2 F}{\partial w_j \partial w_k} = \sum_{m=1}^{M} \sum_{i=1}^{N^o} \left\{ \frac{\partial e_i^{(m)}}{\partial w_k} \frac{\partial e_i^{(m)}}{\partial w_j} + e_i^{(m)} \frac{\partial^2 e_i^{(m)}}{\partial w_j \partial w_k} \right\} \tag{6.7}$$

$$= \sum_{m=1}^{M} \left\{ \boldsymbol{J}^{(m)^T} \boldsymbol{J}^{(m)} + 2\boldsymbol{S}^{(m)} \right\}. \tag{6.8}$$

If we assume $\boldsymbol{S}^{(m)}$ to be small (at the exact solution $\boldsymbol{S}^{(m)}$ equals zero), the Hessian $\boldsymbol{H}$ can be approximated by $\tilde{\boldsymbol{H}}$ where the term $\boldsymbol{S}^{(m)}$ is neglected:

$$\tilde{\boldsymbol{H}} = \sum_{m=1}^{M} \boldsymbol{J}^{(m)^T} \boldsymbol{J}^{(m)}. \tag{6.9}$$

As a result, the update of the parameters in the Levenberg-Marquardt iteration can be computed as

$$\Delta \boldsymbol{w}^{(k)} = -\left\{ \tilde{\boldsymbol{H}} + \mu_k \boldsymbol{I} \right\} \boldsymbol{G}, \tag{6.10}$$

where μ_k is a scalar parameter. Starting with a small value of $\mu_k = 0.01$ scaled with the maximum diagonal element of the hessian, the procedure is almost a Newton iteration. If in the iteration step k no decrease of the objective function is obtained, μ_k is doubled and the iteration step is repeated, otherwise μ_k is divided by two. For large values of μ_k, the iteration is almost a gradient descent method with step length $1/(\mu_k)$. In this work, the Levenberg-Marquardt algorithm is only used for the standard neural networks. This is due to the fact that it does not necessarily converge to a local minima, but might also converge to a saddle point with small negative eigenvalues of the hessian. For the standard neural network, this poses not problem, whereas for the Bayesian network a Laplace approximation is made, where it is required that all eigenvalues of the hessian are strictly positive.

6.2.3.2 Conjugate gradient method - linear case

The conjugate gradient method, developed by [217, 218, 219, 220] solves an optimization problem

$$F(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x} \longrightarrow \min \tag{6.11}$$

with A being a symmetric positive definite matrix, which actually corresponds to the solution of the system of linear equations

$$\frac{F(\boldsymbol{x})}{\partial \boldsymbol{x}} = -\boldsymbol{r}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}. \tag{6.12}$$

At the minimum $\hat{\boldsymbol{x}}$ holds $\boldsymbol{r}(\hat{\boldsymbol{x}}) = \boldsymbol{0}$. Starting from the origin, an iterative solution is obtained by taking a step in the search direction $\boldsymbol{p}^{(i)}$. In steepest descend, the search directions is the negative of the gradient, which often results in a zigzag path to the optimum, since successive search directions might be similar. In contrast, the conjugate gradient approach defines a set of orthogonal directions with respect to $\boldsymbol{A}$, which implies that the search space at iteration k is $\boldsymbol{A}$-orthogonal to the search space $\mathcal{D}_{k-1} = \text{span}\left\{\boldsymbol{p}^{(1)}, .., \boldsymbol{p}^{(k-1)}\right\}$

$$\boldsymbol{p}^{(i)^T}\boldsymbol{A}\boldsymbol{p}^{(j)} = 0 \; \forall i \neq j. \tag{6.13}$$

Assuming for now that this $\boldsymbol{A}$-orthogonal basis is given. Consequently, the solution vector can be expressed as a linear combination of these basis vectors

$$\hat{\boldsymbol{x}} = \sum_{i=1}^{D} \alpha_i \boldsymbol{p}^{(i)}, \tag{6.14}$$

where $\hat{\boldsymbol{x}}$ is the exact solution and D the dimension of the full search space. In an iterative way, the solution can be determined by

$$\boldsymbol{x}^{(i+1)} = \boldsymbol{x}^{(i)} + \alpha_i \boldsymbol{p}^{(i)} \tag{6.15}$$

with $\boldsymbol{x}^{(0)} = \boldsymbol{0}$. In practice, much less iterations are required to obtain an approximate solution within a certain prescribed accuracy range. The error between the exact solution and the approximation at iteration i is $\boldsymbol{A}$-orthogonal to the search direction at iteration i, because the error can be expressed as a linear combination of $\boldsymbol{p}^{(i+1)}, .., \boldsymbol{p}^{(D)}$, which are $\boldsymbol{A}$-orthogonal to $\boldsymbol{p}^{(i)}$. This condition yields an equation for the determination of the coefficients α_i

$$0 = \left(\hat{\boldsymbol{x}} - \boldsymbol{x}^{(i+1)}\right)^T \boldsymbol{A}\boldsymbol{p}^{(i)} \tag{6.16}$$

$$= \left(\hat{\boldsymbol{x}} - \left(\boldsymbol{x}^{(i)} + \alpha_i \boldsymbol{p}^{(i)}\right)\right)^T \boldsymbol{A}\boldsymbol{p}^{(i)}. \tag{6.17}$$

Rearranging finally yields

$$\alpha_i = \frac{\left(\hat{\boldsymbol{x}} - \boldsymbol{x}^{(i)}\right)^T \boldsymbol{A}\boldsymbol{p}^{(i)}}{\boldsymbol{p}^{(i)^T}\boldsymbol{A}\boldsymbol{p}^{(i)}} = \frac{\left(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^{(i)}\right)^T \boldsymbol{p}^{(i)}}{\boldsymbol{p}^{(i)^T}\boldsymbol{A}\boldsymbol{p}^{(i)}} = \frac{\boldsymbol{r}^{(i)^T}\boldsymbol{p}^{(i)}}{\boldsymbol{p}^{(i)^T}\boldsymbol{A}\boldsymbol{p}^{(i)}}. \tag{6.18}$$

The only problem is the determination of the conjugate directions. A straightforward approach would be the steepest descend direction as the initial search direction. The following search directions are then created from the current steepest descend direction by a Gram-Schmidt orthogonalization procedure using all the previous search directions:

$$\boldsymbol{p}^{(k)} = \boldsymbol{r}^{(k)} + \sum_{i=1}^{k-1} \beta_i \boldsymbol{p}^{(i)} \tag{6.19}$$

$$\beta_i = -\frac{{\boldsymbol{r}^{(k)}}^T \boldsymbol{A}\boldsymbol{p}^{(i)}}{{\boldsymbol{p}^{(i)}}^T \boldsymbol{A}\boldsymbol{p}^{(i)}}. \tag{6.20}$$

For practical applications, this approach is not useful, since it requires the storage of all previous gradients and the orthogonalization procedure actually corresponds to a Gauss elimination of the full matrix. However, the storage of all previous search directions and the orthogonalization procedure can be simplified. From Eqs. (6.12), (6.18) and (6.15) it follows that

$$\boldsymbol{r}^{(i+1)} = \boldsymbol{r}^{(i)} - \alpha_i \boldsymbol{A}\boldsymbol{p}^{(i)}, \tag{6.21}$$

with $\boldsymbol{r}^{(i)} = \boldsymbol{r}(\boldsymbol{x}^{(i)})$. Multiplication with ${\boldsymbol{r}^{(k)}}^T$ and rearranging gives

$${\boldsymbol{r}^{(k)}}^T \boldsymbol{A}\boldsymbol{p}^{(i)} = \frac{1}{\alpha_i}\left({\boldsymbol{r}^{(k)}}^T \boldsymbol{r}^{(i)} - {\boldsymbol{r}^{(k)}}^T \boldsymbol{r}^{(i+1)}\right). \tag{6.22}$$

The search space at iteration k is given by $\mathcal{D}^{(k)} = \text{span}\left\{\boldsymbol{p}^{(1)}, .., \boldsymbol{p}^{(k)}\right\}$. Since the residual $\boldsymbol{r}^{(k)}$ is a linear combination of the search directions $\boldsymbol{p}^{(k+1)}, .., \boldsymbol{p}^{(D)}$, which are orthogonal to $\mathcal{D}^{(k)}$ the residual $\boldsymbol{r}^{(k)}$ is also orthogonal to $\mathcal{D}^{(k)}$. Because the search vectors are build by a Gram-Schmidt orthogonalization procedure from the residuals, the space span $\left\{\boldsymbol{r}^{(1)}, .., \boldsymbol{r}^{(k)}\right\}$ is equal to $\mathcal{D}^{(k)}$ and, consequently,

$${\boldsymbol{r}^{(i)}}^T \boldsymbol{r}^{(j)} = \delta_{ij} \tag{6.23}$$

with the Kronecker delta δ_{ij}. Substitution of Eq. (6.23) into Eq. (6.22) gives

$${\boldsymbol{r}^{(k)}} \boldsymbol{A}\boldsymbol{p}^{(i)} = \begin{cases} \dfrac{1}{\alpha_k}{\boldsymbol{r}^{(k)}}^T \boldsymbol{r}^{(k)} & k = i \\ \dfrac{1}{\alpha_{k-1}}{\boldsymbol{r}^{(k)}}^T \boldsymbol{r}^{(k)} & k = i+1 \\ 0 & \text{otherwise.} \end{cases} \tag{6.24}$$

Comparing this result with Eq. (6.20), it is realized that almost all β_i vanish and Eq. (6.19) can be simplified to

$$\boldsymbol{p}^{(k)} = \boldsymbol{r}^{(k)} - \frac{{\boldsymbol{r}^{(k)}}^T \boldsymbol{r}^{(k)}}{\alpha_{k-1}{\boldsymbol{p}^{(k-1)}}^T \boldsymbol{A}\boldsymbol{p}^{(k-1)}}\boldsymbol{p}^{(k-1)} \tag{6.25}$$

$$= \boldsymbol{r}^{(k)} - \frac{{\boldsymbol{r}^{(k)}}^T \boldsymbol{r}^{(k)}}{{\boldsymbol{r}^{(k-1)}}^T \boldsymbol{r}^{(k-1)}}\boldsymbol{p}^{(k-1)} \tag{6.26}$$

In summary, only the previous residual and the previous search direction have to be stored and used for the orthogonalization procedure, which decreases the effort from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$.

Another drawback of the above procedure is the requirement to compute and store the Hessian matrix, which is used for the calculation of the parameters α_i in Eq. (6.18). Fortunately, there is a possibility to apply the method without the need for the Hessian matrix. Assume that we start at a point $\boldsymbol{x}^{(i)}$, and perform a line search along the direction $\boldsymbol{p}^{(i)}$, such that $F(\boldsymbol{x}^{(i+1)})$ has a minimum at λ with $\boldsymbol{x}^{(i+1)} = \boldsymbol{x}^{(i)} + \lambda \boldsymbol{p}^{(i)}$. Consequently, the following relation holds:

$$0 = \left(\frac{\partial F(\boldsymbol{x}^{(i+1)})}{\partial \lambda_i} \right)^T \boldsymbol{p}^{(i)} = -\boldsymbol{r}^{(i)^T} \boldsymbol{p}^{(i+1)}. \tag{6.27}$$

Similar to Eq. (6.21), it follows that

$$\boldsymbol{r}^{(i+1)} = \boldsymbol{r}^{(i)} - \lambda_i \boldsymbol{A} \boldsymbol{p}^{(i)}. \tag{6.28}$$

Multiplication with $\boldsymbol{p}^{(i)^T}$, using Eq. (6.27) and rearranging finally gives

$$\lambda_i = \frac{\boldsymbol{p}^{(i)^T} \boldsymbol{r}^{(i)}}{\boldsymbol{p}^{(i)^T} \boldsymbol{A} \boldsymbol{p}^{(i)}}, \tag{6.29}$$

which is identical to Eq. (6.18) and, consequently, $\lambda_i = \alpha_i$. It is to be noted that λ_i was calculated without the knowledge of the Hessian matrix $\boldsymbol{A}$, but a line search was used instead. This result is especially useful in the nonlinear case.

6.2.3.3 Conjugate gradient method - nonlinear case

The problem of determining the weights and biases in the training procedure of the neural networks is a highly nonlinear problem, with a Hessian matrix that is not positive definite and several local minima. This can already be realized by changing the order of the neurons in the hidden layer and the corresponding weights in a network with a single hidden layer. The response of the network does not change, which implies that at least $N^h!$ global minima exist, where N^h is the number of neurons in the hidden layer. However, it is often sufficient to find a local minima of the objective function. In the vicinity of a local minima, the objective function can accurately be approximated by a quadratic function with a positive definite Hessian matrix. The main problem in the nonlinear case is the calculation of the parameters α_i. If the exact Hessian matrix cannot be calculated, e.g. if it is computationally expensive as in the case of neural networks, a linesearch in the search direction $\boldsymbol{p}^{(i)}$ can be performed as elucidated in the previous section. If the quadratic approximation of the objective function is exact, this approach is identical to the conjugate gradient method in the linear case. For an objective function that deviates from the exact quadratic function, the linesearch assures that a decrease of the objective function is obtained. For the nonlinear case Polak and Ribière modified the update of the new search directions [220]:

$$\boldsymbol{p}^{(k)} = \boldsymbol{r}^{(k)} - \frac{\left(\boldsymbol{r}^{(k)} - \boldsymbol{r}^{(k-1)}\right)^T \boldsymbol{r}^{(k)}}{\boldsymbol{r}^{(k-1)^T} \boldsymbol{r}^{(k-1)}} \boldsymbol{p}^{(k-1)}. \tag{6.30}$$

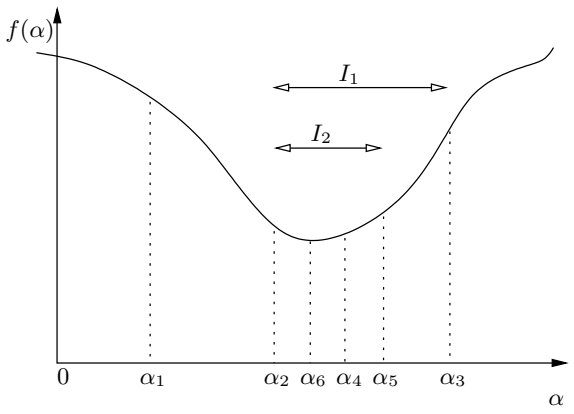


Figure 6.2 Line search based on the golden section search.

In the linear case, there is no difference between Eq. (6.30) and Eq. (6.26), since $\boldsymbol{r}^{(k-1)^T}\boldsymbol{r}^{(k)} = 0$ according to Eq. (6.23). In the nonlinear case, Eq. (6.30) seems to outperform the original Fletcher-Reeves formula [221]. Due to numerical inaccuracies and nonlinearities of the objective function, a restart after a certain number of iterations is required, i.e. setting the search direction to the steepest descent direction. In this work, a criterion based on [222] has been implemented. A restart is performed, if the inequality

$$|\boldsymbol{r}^{(k)^T}\boldsymbol{r}^{(k-1)}| \geq 0.2|\boldsymbol{r}^{(k)^T}\boldsymbol{r}^{(k)}| \tag{6.31}$$

is fulfilled, which requires that successive gradients of the objective function are not colinear up to a certain degree described by the factor 0.2. Under the assumption of $\|\boldsymbol{r}^{(k)}\| = \|\boldsymbol{r}^{(k+1)}\|$, Eq. (6.31) can be interpreted as requiring a minimum angle between successive gradients of the objective function of 78.5^o.

As highlighted above, the computational expensive part in the training of neural networks with the conjugate gradient approach is the efficient implementation of a line search. For this reason, different line searches have been investigated. Each line search method can be decomposed into two steps. In the first step, an interval is determined, in which the potential minimum is located, and in a second step, the interval is narrowed until a prescribed accuracy is achieved. In the following, two implemented line searches are presented and analyzed with respect to their computational speed and robustness.

Golden section search

A robust linesearch is the Golden section search, which requires only function evaluations. The main idea is illustrated in Fig. 6.2. At first, an interval is determined with $f(\alpha_i) > f(\alpha_{i+1})$ and $f(\alpha_{i+2}) > f(\alpha_{i+1})$. Starting from the linesearch parameter $\alpha_0 = 0$ and an initial interval $\Delta\alpha$, the function is evaluated at $\alpha_1 = \alpha_0 + \Delta\alpha$. If $f(\alpha_1)$ is smaller than $f(\alpha_0)$, the procedure is repeated until $f(\alpha_i) > f(\alpha_{i-1})$, with $\alpha_i = \alpha_{i-1} + \Delta\alpha$. Note that in each iteration the size $\Delta\alpha$ is multiplied by a factor >1, in order to increase the search interval. In the case $f(\alpha_0) < f(\alpha_1)$, the minimum is inside the interval $[\alpha_0, \alpha_1]$ and a bisection algorithm is used until $f(\alpha_i) < f(\alpha_0)$. In Fig. 6.2, the first interval I_1 which comprises the minimum is $[\alpha_2, \alpha_3]$.

Within this interval, two additional points α_4 and α_5 are determined.

$$s = \frac{-1+\sqrt{5}}{1+\sqrt{5}} (\alpha_b - \alpha_a) \tag{6.32}$$

$$\alpha_{ai} = \alpha_a + s \tag{6.33}$$

$$\alpha_{bi} = \alpha_b - s, \tag{6.34}$$

where the left border of the interval is labeled α_a, the right border α_b and the intermediate points α_{ai} and α_{bi}. If $f(\alpha_{ai}) < f(\alpha_{bi})$, the procedure is repeated within the interval $[\alpha_a, \alpha_{bi}]$, otherwise the procedure is repeated within the interval $[\alpha_{ai}, \alpha_b]$. It is to be noted that the position of the intermediate points is calculated from the golden ratio. This has the advantage that for a new interval only one additional function value for the intermediate points has to be calculated, whereas the other one can be recycled from the previous interval. In Fig. 6.2, the second interval is given by $[\alpha_2) < \alpha_5]$, and only $f(\alpha_6)$ has to be calculated, whereas $f(\alpha_4)$ can be used from the previous interval. The procedure is repeated until the size of the interval $\alpha_b - \alpha_a$ is smaller than a user-defined tolerance.

Brent's method

The golden section search is a robust line search, but it requires a lot of function evaluations until the minimum is bracketed. For a pure quadratic function is seems to be reasonable to approximate the function by a quadratic function using the intermediate point and the borders of the interval. The minimum of this quadratic function can then be used as a new intermediate point. However, a closer look at this approach reveals that the algorithm is not robust, it might get stuck in certain points or oscillate around the exact solution. In order to circumvent these problems, a method coupling the advantage of a quadratic interpolation and the golden section search is used [221, 223]. The principal idea is that the quadratic approximation converges faster to the minimum, whereas in cases, where the quadratic approximation gets stuck, the golden section search is used.

The extremum of a quadratic function defined at the points a, b and c is given by

$$x = b - \frac{1}{2} \frac{(b-a)^2 (f(b) - f(c)) - (b-c)^2 (f(b) - f(a))}{(b-a)(f(b) - f(c)) - (b-c)(f(b) - f(a))}. \tag{6.35}$$

The algorithm is schematically summarized in Algorithm 6.1. It stores 5 locations and the corresponding function values - x corresponds to the smallest function value observed, w corresponds to the point with the second smallest function value and v is the previous value of w. The current search interval is described by a and b. Compared to the standard golden section search, this line search could reduce the number of function evaluations required to achieve the same accuracy by a factor of approximately three. Since the line search is the computational expensive part of the training procedure using conjugate gradients, a substantial decrease of the computational time could be obtained.

Algorithm 6.1 Brent's method as a line search combining a quadratic approximation with a golden section search

Calculate the left and right borders a and b and an intermediate value x
  with $f(a) > f(x)$ and $f(b) > f(x)$
Initialize w=v=x and step length of step before previous $e = 0$
while Interval length $b - a >$ tolerance **do**
  if $\|e\| >$ tolerance (the step before the previous is not negligible) **then**
    Calculate the minimum x_{min} according to Eq. (6.35)
    if x_{min} is inside the interval $[a, b]$ and a sufficiently away from the current minima a **then**
      update step length $e = d$
      accept the parabolic step with step length $d = x_{min} - x$
    else
      Take golden section step into the larger segment $[a, x]$ or $[x, b]$
      if $x > \frac{a+b}{2}$ **then**
        Update step length $e = a - x$
      else
        Update step length $e = b - x$
      $d = 0.38196e$
  else
    Take a golden section step into the larger segment $[a, x]$ or $[x, b]$
  Evaluate function at $u = x + d$
  if $f(u) < f(x)$ (smaller function value) **then**
    if $u \geq x$ **then**
      Update left border $a = x$
    else
      Update right border $b = x$
    Update variables $v = w$, $w = x$, $x = u$
  else
    if $u < x$ **then**
      Update left border $a = u$
    else
      Update right border $b = u$
    if $f(u) \leq f(w)$ or $w = x$ **then**
      Update variables $v = w$, $w = u$
    else
      if $f(u) \leq f(v)$ or $v = x$ or $v = w$ **then**
        Update variables $v = u$

Preconditioning

The conjugate gradient method works efficiently, if the objective function corresponds to a hypersphere, i.e. the Hessian matrix $\boldsymbol{A}$ is identical to the unity matrix. A measure often used to describe the deviation from the sphere is the condition number κ

$$\kappa = \|\boldsymbol{A}\| \left\|\boldsymbol{A}^{-1}\right\| = \frac{\lambda_{max}}{\lambda_{min}}, \tag{6.36}$$

where $\lambda_{max}/\lambda_{min}$ is the largest/smallest eigenvalue of the symmetric matrix $\boldsymbol{A}$ and $\|\cdot\|$ in Eq. (6.36) is the L_2-norm. For a sphere, the condition number is unity and a small error on the right hand side induces a small error on the solution vector. In contrast, if the condition number is large, a small error in the right hand side may cause a large error in the solution vector. In order to avoid an ill-conditioned system, a preconditioner should be added in the solution procedure. This is especially useful for Bayesian neural networks discussed in the following section, because large differences in magnitude between the hyperparameters α_i may lead to ill-posed problems with a condition number larger than 10^{10}.

Two principal preconditioning methods to transform Eq. (6.12) in a well-posed problem can be distinguished - a right preconditioner

$$\boldsymbol{A}\boldsymbol{P}^{-1}\boldsymbol{P}\boldsymbol{x} = \boldsymbol{b} \tag{6.37}$$

or the left preconditioner

$$\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{P}^{-1}\boldsymbol{b} \tag{6.38}$$

with a preconditioning matrix $\boldsymbol{P}$, which is easy to invert. In this implementation, a combination of both approaches using a Jacobi preconditioner has been used, where the matrix $\boldsymbol{P}$ is given by the diagonal elements of the Hessian matrix $\boldsymbol{A}$. In order to avoid numerical instabilities due to small or negative diagonal elements, the standard Jacobi preconditioner has been modified and the matrix $\boldsymbol{P}$ is defined as

$$P_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j \text{ and } A_{ii} < 1 \\ A_{ii} & \text{otherwise} \end{cases} \tag{6.39}$$

In the nonlinear case, this procedure actually corresponds to the transformation of the original search space $\boldsymbol{x}$ to the transformed space $\boldsymbol{y} = \boldsymbol{P}\boldsymbol{x}$ and a local approximation with a hypersphere. It is to be noted that the only difference in the prescaled version of the conjugate gradient method is the calculation of the gradients, which are given by

$$\boldsymbol{g}_{pre} = \frac{\partial F}{\partial \boldsymbol{y}} \tag{6.40}$$

$$= \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{y}} \frac{\partial F}{\partial \boldsymbol{x}} = \boldsymbol{P}^{-1}\boldsymbol{g}_{orig}, \tag{6.41}$$

where $\boldsymbol{g}_{orig}$ and $\boldsymbol{g}_{pre}$ are the gradients in the original and the preconditioned space. Furthermore, the linesearch has to be performed in the preconditioned space as well. However, since only a scalar step length is calculated, the search direction in the original space $\boldsymbol{p}_{orig}^{(k)}$ can be calculated from the preconditioned search direction $\boldsymbol{p}_{pre}^{(k)}$ a priori and then the linesearch is performed in the original system without the need for the transformation in each linesearch step.

$$\boldsymbol{p}_{orig}^{(k)} = \boldsymbol{P}^{-1}\boldsymbol{p}_{pre}^{(k)}. \tag{6.42}$$

In the case of neural networks, the Hessian matrix is not constant. Consequently, the approximation using only the Hessian at the starting point of the conjugate gradient iteration may be considerably different at subsequent iterations. In order to circumvent this problem, a preconditioning step at each reset of the search direction is used.

The implemented algorithm for the training procedure using conjugate gradients is summarized in Algorithm 6.2.

6.2.4 Numerical implementation

6.2.4.1 Transformation of input and output data

In order to improve the training procedure, several heuristics are applied. One of these procedures is the scaling of the input and the output data to have a zero mean and standard deviation of one [224]. This is performed for all variables separately by calculating the mean μ and the standard deviation σ of this variable for all training samples, and then performing a transformation according to

$$\tilde{x} = \frac{x - \mu}{\sigma}. \tag{6.43}$$

The training of the network is then performed in the transformed space $\tilde{x}$. A further improvement, not implemented in this work, can be obtained, when the correlation between the input and the correlation between the output variables is removed.

6.2.4.2 Initialization of weights

The optimization strategies described in the previous section start at an initial point. Initializing all weights and biases in the network with a constant factor leads to a highly ill-conditioned problem. Since due to symmetry reasons all neurons and the corresponding weights in a single layer have the same influence on the result, the gradient of the objective function with respect to these weights is similar. As a result, these weights will only change in a similar way during the optimization procedure. In order to circumvent this problem, the weight w_{ij} connecting neuron i with neuron j is initialized in a stochastic procedure according to [224] from a uniform distribution in the interval $[-s_{ij}, s_{ij}]$ with

$$s_{ij} = \sqrt{\frac{3}{L_{ij}}}. \tag{6.44}$$

Algorithm 6.2 Preconditioned conjugate gradient method used for the training of neural networks

Initialize the starting point $\boldsymbol{x}^0$ and the previous gradient $\boldsymbol{r}_0 = \boldsymbol{0}$
Calculate the scale factors $\boldsymbol{s}$ for preconditioning from the Hessian matrix $\boldsymbol{A}$

$$s_i = \begin{cases} 1 & A_{ii} < 1 \\ A_{ii} & \text{otherwise} \end{cases}$$

while $\|\boldsymbol{r}_{pred}^{(i)}\| < \text{tol}$ **do**
Calculate the negative gradient of the objective function $\boldsymbol{r}_{orig}^{(i)}$ at $\boldsymbol{x}^{(i)}$
Precondition $\left(\boldsymbol{r}_{pre}^{(i)}\right)_j = \dfrac{\left(\boldsymbol{r}_{orig}^{(i)}\right)_j}{s_j}$
Check for a restart using the Powell-Beale formula
if $|\boldsymbol{r}_{pre}^{(k)^T} \boldsymbol{r}_{pre}^{(k-1)}| \geq 0.2 |\boldsymbol{r}_{pre}^{(k)^T} \boldsymbol{r}_{pre}^{(k)}|$ **then**
Calculate the scale factors $\boldsymbol{s}$ for preconditioning from the Hessian matrix $\boldsymbol{A}$

$$s_i = \begin{cases} 1 & A_{ii} < 1 \\ A_{ii} & \text{otherwise} \end{cases}$$

Initialize the search direction $\boldsymbol{p}_i^{pre}$ with the negative gradient in the original space $\boldsymbol{r}_i^{orig}$ preconditioned with $\boldsymbol{s}$

$$p_{pre}^{(i)} = -\frac{r_{orig}^{(i)}}{s_i}$$

else
Calculate the new search direction $\boldsymbol{p}_{pre}^{(i)}$ using the Polak-Ribière update formula

$$\boldsymbol{p}^{(i)} = \boldsymbol{r}^{(i)} - \frac{\left(\boldsymbol{r}^{(i)} - \boldsymbol{r}^{(i-1)}\right)^T \boldsymbol{r}^{(i)}}{\boldsymbol{r}^{(i-1)^T} \boldsymbol{r}^{(i-1)}} \boldsymbol{p}^{(i-1)}$$

Transform the search direction into the original space

$$p_{orig}^{(i)} = \frac{p_{pre}^{(i)}}{s_i}$$

Perform a linesearch using Brent's method along the search direction $p_{orig}^{(i)}$ described in Algorithm 6.1 and calculate the new state $\boldsymbol{x}^{(i+1)}$

L_{ij} is the number of input connections of the neuron. The initial bias values are all set to zero. For a comparison of different weight initialization procedures see e.g. [225].

6.2.4.3 Regularization

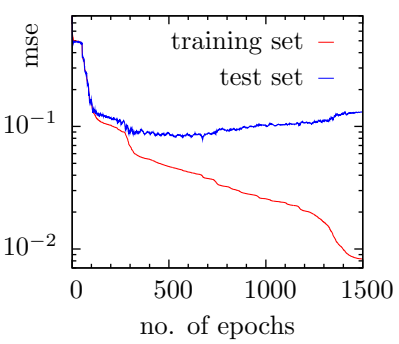
For practical applications, the complexity of the function to be approximated is not known in advance. Consequently, the number of neurons and hidden layers required to approximate the function is difficult to determine a priori. On the one hand, the network should be as complex as necessary in order to accurately capture all the essential features of the exact function, whereas, on the other hand, the network should be as simple as possible to avoid overfitting, i.e. obtain a network that generalizes well. Obviously, not only the function to be approximated has in influence on the optimal network architecture (number of neurons and number of hidden layer), but also the number of training samples. Having only a few training samples, the simple network with only a few neurons in the hidden layer is sufficient, whereas for many training samples the number of neurons in the hidden layer can be increased.

In order to solve these problems, different approaches are proposed in the literature. In general, they can be subdivided into two classes. In the first class of methods, the dimension of the parameter space is adapted to the problem. Examples of these methods are network pruning [226, 227, 228], where weights are incrementally removed from the network, or the application of cascade-correlation network architectures [229], where - starting with a simple network - hidden units are incrementally added. In the second class of methods, the size of the parameters is reduced. Examples for this approach are regularization techniques such as weight decay [230] or early stopping [231]. In the weight decay approach, an additional term besides the mean square error is added to the objective function which corresponds to the norm of the weight vector multiplied by a user-defined regularization parameter, whereas in the early stopping approach the training of the network is not performed until convergence but stopped earlier based on a criterion using a test set. It is to be noted that Bayesian neural networks discussed in the next section are also based on the regularization approach, but the user-defined regularization parameters are automatically determined.

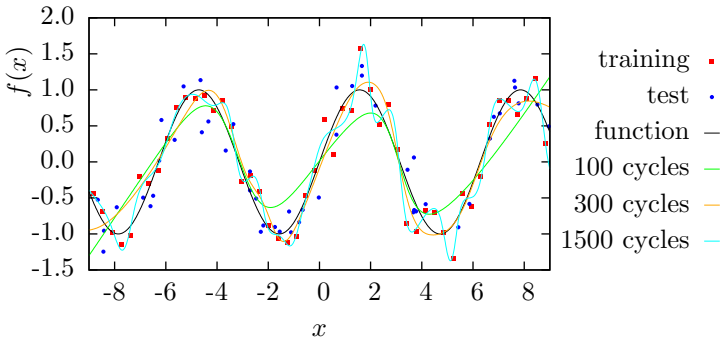
Two of these methods have been implemented and, in the following, are exemplarily illustrated.

Early stopping

The idea is based on the fact that during the training procedure the network captures first the general trend and then adapts to local phenomena. For that purpose, the set of training samples is split into a test and a training set. The training set is used in the training of the network, whereas after each epoch the objective function of the test set is evaluated. A typical curve for the error in the test and training set is illustrated in Fig. 6.3 using an example in 1-D with 50 training samples and 50 test samples. The function to be approximated is $f(x) = sin(x)$ in the interval [-9,9]. The training and test samples are created using latin hypercube sampling. Additional Gaussian noise is added to the function values with a standard deviation of 0.25 and a multilayer perceptron with a single hidden layer and 100 neurons in the hidden layer has been used.



(a) Mean and standard deviation of the mean square error for the test and the training set.



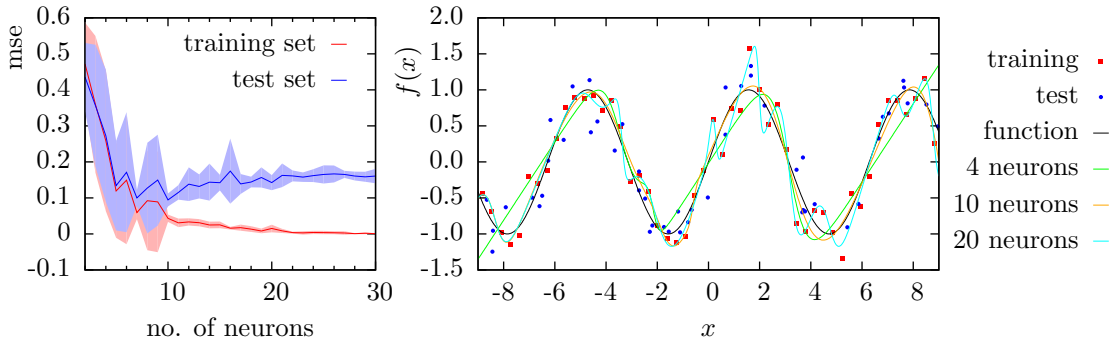
(b) Influence of the number of epochs on the approximation.

Figure 6.3 Illustration of the early stopping approach to train neural networks using a preconditioned conjugate gradient algorithm.

At first, the mean square error of the test and the training sets decreases. At a certain number of epochs, the mean square error of the test set does not decrease any more or even increases. This is the point where to stop the training procedure. However, this procedure has several drawbacks. The first problem is the requirement of a test set, which implies that only a subset of the numerical data is actually used in the training procedure. The second problem is to define a criteria when to stop. As can be verified from Fig. 6.3, the mean square error of the test set is not a smooth curve which decreases and, at a certain point, increases, but it is more or less oscillating around this curve. In order to apply this method, a smoothing approach over a certain number of iterations is required, but the influence radius (number of iterations used in the smoothing procedure to determine a smoothed value) is problem dependend and must be defined by the user. For a detailed discussion on several stopping criteria see e.g. [231].

Adaptive complexity of the network

Another approach is based on limiting the complexity of the network. Similar to the previous approach, the total set is decomposed into a test and a training set. Starting with the simplest network with a single neuron in the hidden layer, the network is trained until convergence (e.g. norm of gradient of the objective function smaller than a certain value) and the error for the test and training set is evaluated. The procedure is repeated with the number of neurons in the hidden layer increased by a certain value. A curve obtained by this procedure is exemplarily illustrated in Fig. 6.4(a). It is well known that due to the stochastic initialization of the network parameters, the final response of the network differs even for the same training data. Consequently, the procedure is repeated several times (in this example ten times) with a constant number of neurons. Afterwards, the mean and the standard deviation of the mean square error can be calculated for the test and the training set. In general, a minimum of the mean square error of the test set can be identified. In the current example, this optimal complexity of the network is approximately 10 neurons in the hidden layer. Afterwards, the network is retrained with the number of neurons in the hidden layer fixed to 10, but this time



(a) mean square error (b) approximation

Figure 6.4 Development of the mean square error during a run of a training procedure for the training and an additional test set using a preconditioned conjugate gradient algorithm.

using the total set of data. Analog to the previous remarks concerning the influence of the stochastic character of the initialization procedure it might be advantageous to approximate the output by the mean of several networks trained using different initial weights and biases.

The influence of the number of neurons in the hidden layer on the network approximation is illustrated in Fig. 6.4(b). For a simple network with only 4 neurons, the complexity of the network is not sufficient to capture certain local curvatures, but the global trend is already accurately reproduced. An excellent generalization behavior exhibits the network with 10 neurons in the hidden layer, while better reproducing local effects. The network with 20 neurons in the hidden layer clearly shows overfitting characteristics and the generalization capacity of the network is only limited. One of the main drawbacks of this approach is the computational complexity of the procedure. Since for each number of neurons in the hidden layer the network has to be trained several times to account for the stochastic character of the initialization, a large computational effort is required. Furthermore, overfitting is only reduced by determining the simplest network architecture that is able to approximate the training data, but cannot be eliminated. This is due to the fact that depending on the initial weights, the network trained with the full set might still overfit in certain regions. Another disadvantage is the user-defined parameter describing the ratio between the size of the test and the training set. In this context, generalized cross validation [232] has been used. The idea is to split the total set of training samples into a certain number of sets. For each set, the neural network is trained using all other sets but the one, and afterwards the mean square error of the test set is evaluated for this set. The average of the mean square error over all sets can be used similar to the mean square error in the test set to identify overfitting.

6.3 Bayesian neural networks

6.3.1 Motivation

In contrast to the frequentist approach demonstrated in the previous section, where the weights and biases after the training of the neural network are fixed and the output for a given input is deterministic, the Bayesian approach interprets the free parameters and the output of the network as stochastic variables, where the mean values are related to the fixed parameters in the frequentist approach. This offers the advantage of estimating not only the input/output relation, but additionally evaluate the quality of the approximation. Furthermore, a regularization procedure is automatically included in the training procedure. Comprehensive reviews of Bayesian methods can be found in [233, 234].

In order to illustrate the Bayesian approach, an example from [233] is illustrated in Fig. 6.5. Consider three different models, which all have a single parameter w and a constant output t which is the parameter w disturbed with a certain amount of noise. For all three models, prior knowledge about the size of the parameter w exists. In the first model, the interval is rather large, whereas in the third model the parameter w is located only in a narrow interval. For all these models, data points t are generated by sampling w with the prior distribution and then adding the noise. These sampling points are the clouds of points in Fig. 6.5. Consequently, a probability density function $p(t|\mathcal{H}_i)$ for all three models can be calculated. Assume furthermore, that an additional point D is given. The question is, which model has the highest probability that datapoint D has been created by that model. The probability that datapoint D has been created by $\mathcal{H}_3$ is very small, since the prior distribution of w is very narrow. For model $\mathcal{H}_1$, the probability that datapoint D has been created by this model is also small, but this time due to the fact that the probability density function of the response t is widely spread and, consequently, smaller at a certain fixed point (since the integral over the total domain is 1). The most probable model is consequently model $\mathcal{H}_2$. In the following derivation for Bayesian training of neural networks, the width of the prior distributions is characterized by the hyperparameters $\boldsymbol{\alpha}$, the noise in the model is represented by the noise covariance $\boldsymbol{\Sigma}_\beta$ and the weights and biases are the free parameters $\boldsymbol{w}$ of the model.

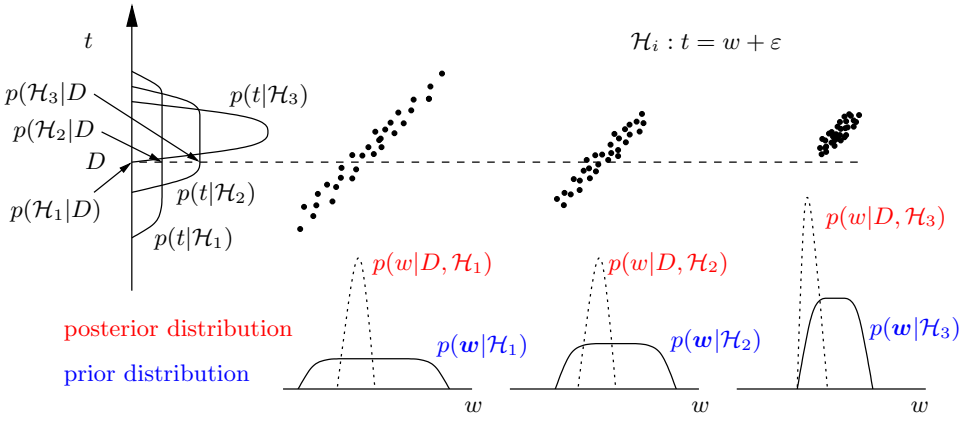


Figure 6.5 Model selection using a Bayesian approach according to [233].

6.3.2 Architecture

Within the Bayesian framework of neural network interpolation, the output of the neural network $\boldsymbol{y}$ given the input $\boldsymbol{x}$ is assumed to be superposed by Gaussian noise:

$$p(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{\Sigma}_\beta) = \mathcal{N}(\boldsymbol{y}(\boldsymbol{x}, \boldsymbol{w}), \boldsymbol{\Sigma}_\beta), \tag{6.45}$$

where $\mathcal{N}(\boldsymbol{y}(\boldsymbol{x}, \boldsymbol{w}), \boldsymbol{\Sigma}_\beta)$ is a standard joint Gaussian distribution with mean value $\boldsymbol{y}(\boldsymbol{x}, \boldsymbol{w})$ and covariance $\boldsymbol{\Sigma}_\beta$. It is to be noted that, in contrast to standard Bayesian neural networks, the author uses a full covariance matrix $\boldsymbol{\Sigma}_\beta$, with a separate noise variance for each output of the network to account for a different noise level in each output component and their mutual correlation. In a similar way, the prior distribution of the free parameters (weights and biases), which corresponds to the distribution of the free parameters without any knowledge of the training data $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{t})$, is assumed to be

$$p(w_j|\alpha_j) = \mathcal{N}(w_j|0, \alpha_j^{-1}). \tag{6.46}$$

Theoretically, a precision parameter α_j can be defined for all free parameters. However, the network possesses a symmetry with respect to the neurons (e.g. if neuron 1 and 2 in the hidden layer with their associated weights are exchanged, the input-output relation will remain unchanged). In order to account for this symmetry property, the free parameters are placed into groups with a single precision parameter for each group, which is illustrated in Fig. 6.6 for a network with a single hidden layer and two inputs and two outputs. A group is created for each input neuron with all connections (weights) emanating from that input (groups 1 and 2). Furthermore, an additional group is created for the biases $b_i^{(1)}$ of the first layer (group 5) and, finally, a group for each output with its bias and all connections flowing into that output neuron (groups 3 and 4) is created. If an additional second hidden layer should be introduced, an additional group with the weights connecting both layers and the biases of the second layer has to be added.

6.3.3 Calculation of weights and biases

Assuming furthermore that the hyperparameters α_i and the noise covariance $\boldsymbol{\Sigma}_\beta$ are known, and using the assumption that the samples are independent and identically distributed (i.i.d.), it follows from Eq. (6.45)

$$p(\boldsymbol{T}|\boldsymbol{X}, \boldsymbol{w}, \boldsymbol{\Sigma}_\beta) = \prod_{m=1}^{M} p(\boldsymbol{t}^{(m)}|\boldsymbol{x}^{(m)}, \boldsymbol{w}, \boldsymbol{\Sigma}_\beta), \tag{6.47}$$

where $\boldsymbol{T}$ describes the output of all training samples, $\boldsymbol{t}^{(m)}$ is the output of training sample m and $\boldsymbol{X}$ contains the input vectors $\boldsymbol{x}^{(m)}$ for all training samples. Since the inputs of the training data are fixed, the notation $p(\boldsymbol{T}|\boldsymbol{X})$ is sometimes replaced by $p(\mathcal{D})$, which is the probability that the data is produced by the neural network model.

Marginalizing over the weights, the probability of the data for the given model can be calculated:

$$p(\boldsymbol{T}|\boldsymbol{X}, \boldsymbol{\alpha}, \boldsymbol{\Sigma}_\beta) = \int p(\boldsymbol{T}|\boldsymbol{X}, \boldsymbol{w}, \boldsymbol{\Sigma}_\beta)\, p(\boldsymbol{w}|\boldsymbol{\alpha})\, d\boldsymbol{w}. \tag{6.48}$$

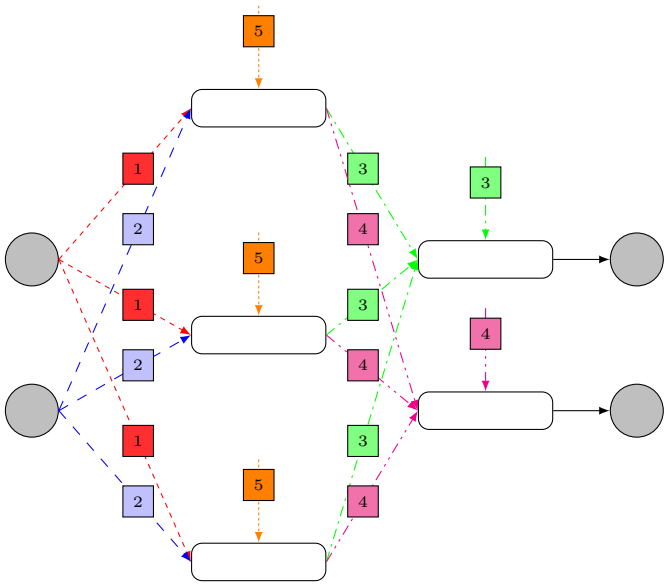


Figure 6.6 Definition of groups for the free parameters (weights and biases).

Using Bayes theorem $p(A|B)\ p(B) = p(B|A)\ p(A)$, it directly follows that

$$p(\boldsymbol{w}|\mathcal{D}, \boldsymbol{\alpha}, \boldsymbol{\Sigma}_\beta) \propto p(\mathcal{D}|\boldsymbol{w}, \boldsymbol{\Sigma}_\beta)\ p(\boldsymbol{w}|\boldsymbol{\alpha}), \tag{6.49}$$

where the factor of proportionality is $p(\mathcal{D}|\boldsymbol{\alpha}, \boldsymbol{\Sigma}_\beta)^{-1}$, which is independent of the free parameters $\boldsymbol{w}$. Due to the nonlinearity of the input-output relation of the neural network, the posterior distribution of the free parameters $p(\boldsymbol{w}|\mathcal{D}, \boldsymbol{\alpha}, \boldsymbol{\Sigma}_\beta)$ is non-Gaussian. However, a Gaussian approximation to the posterior distribution using the Laplace approximation [235] can be determined. When maximizing the logarithm of the posterior distribution of $\boldsymbol{w}$, which is, due to the monotonic character of the ln-function identical to maximizing the posterior distribution itself, the products in Eq. (6.47) are converted to sums and it follows that

$$\ln p(\mathcal{D}|\boldsymbol{w}, \boldsymbol{\Sigma}_\beta) = \sum_{m=1}^{M} \ln \left[\frac{1}{(2\pi)^{\frac{N^o}{2}}} \frac{1}{|\boldsymbol{\Sigma}_\beta|^{\frac{1}{2}}} e^{\left(\boldsymbol{t}^{(m)} - \boldsymbol{y}^{(m)}\right)^T \boldsymbol{\Sigma}_\beta^{-1} \left(\boldsymbol{t}^{(m)} - \boldsymbol{y}^{(m)}\right)} \right] \tag{6.50}$$

$$= \sum_{m=1}^{M} \frac{1}{2} \left[-N^o \ln 2\pi - \ln |\boldsymbol{\Sigma}_\beta| - \left(\boldsymbol{t}^{(m)} - \boldsymbol{y}^{(m)}\right)^T \boldsymbol{\Sigma}_\beta^{-1} \left(\boldsymbol{t}^{(m)} - \boldsymbol{y}^{(m)}\right) \right], \tag{6.51}$$

which gives, using Eqs. (6.49) and (6.46):

$$\ln p(\boldsymbol{w}|\mathcal{D}, \boldsymbol{\alpha}, \boldsymbol{\Sigma}_\beta) = \ln p(\mathcal{D}|\boldsymbol{w}, \boldsymbol{\Sigma}_\beta) + \ln p(\boldsymbol{w}|\boldsymbol{\alpha}) + \text{const} \tag{6.52}$$

$$= -\frac{1}{2} \left[\sum_{m=1}^{M} \left(\boldsymbol{t}^{(m)} - \boldsymbol{y}^{(m)}\right)^T \boldsymbol{\Sigma}_\beta^{-1} \left(\boldsymbol{t}^{(m)} - \boldsymbol{y}^{(m)}\right) + \sum_{j=1}^{N^w} \alpha_j w_j^2 \right] + \text{const}. \tag{6.53}$$

N_o is the number of output parameters, N_w is the number of free parameters and M the number of training samples. The terms independent of $\boldsymbol{w}$ are summed up in the constant term, since they have no influence in the optimization procedure. Comparing Eq. (6.53) to the objective function of standard neural networks in Eq. (6.3) including only the sum of squared errors, it is observed that the noise covariance $\boldsymbol{\Sigma}_\beta$ serves as a scaling parameter for the sum of squared errors, and an additional regularization term is added that minimizes the norm of the free parameters. Minimizing the negative posterior distribution of the free parameters using an appropriate training procedure (e.g. conjugate gradients) gives a mode of the posterior distribution $\boldsymbol{w}_{MAP}$ (maximum à posteriori). Using Eq. (6.53), the covariance matrix $\boldsymbol{A}$ of the gaussian approximation to the posterior distribution at $\boldsymbol{w}_{MAP}$ is given by

$$\boldsymbol{A} = -\nabla\nabla \ln p(\boldsymbol{w}_{MAP}|\mathcal{D}, \boldsymbol{\alpha}, \boldsymbol{\Sigma}_\beta) \tag{6.54}$$

$$= \frac{1}{2}\frac{\partial}{\partial \boldsymbol{w}^2}\left[\sum_{m=1}^{M}\left(\boldsymbol{t}^{(m)} - \boldsymbol{y}^{(m)}\right)^T \boldsymbol{\Sigma}_\beta^{-1}\left(\boldsymbol{t}^{(m)} - \boldsymbol{y}^{(m)}\right)\right] + \operatorname{diag}\{\alpha_1, .., \alpha_{N^w}\} \tag{6.55}$$

The second derivatives of the output with respect to the weights can be either approximated similar to Eq. (6.9) with the scaling parameters $\boldsymbol{\Sigma}_\beta$ added, or exactly calculated using a combined backpropagation and forward propagation scheme as explained in [236]. The advantage of using the approximated hessian is, on the one hand, the computational efficiency, since the exact calculation of the Hessian is computationally much more expensive, and, on the other hand, the property of the approximated Hessian to be positive definite. From a theoretical point of view, the Hessian matrix of $p(\boldsymbol{w}_{MAP}|\mathcal{D}, \boldsymbol{\alpha}, \boldsymbol{\Sigma}_\beta)$ at a local minimum $\boldsymbol{w}_{MAP}$ is also positive definite, but in the following calculations it is also required that the second derivative of the sum of mean square errors be positive definite (see e.g. [237]). This can only be obtained by an eigenvalue analysis of the exact Hessian matrix, setting all negative eigenvalues to a small positive value and back transformation into the original system. In order to circumvent all these problems, the approximated Hessian matrix is used.

The approximation of the posterior distribution of the free parameters can then be calculated by

$$\tilde{p}(\boldsymbol{w}|\mathcal{D}, \boldsymbol{\alpha}, \boldsymbol{\Sigma}_\beta) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_{MAP}, \boldsymbol{A}^{-1}). \tag{6.56}$$

6.3.4 Predictive distribution

The predictive distribution of the output $\boldsymbol{t}$ for a new input vector $\boldsymbol{x}$ is given by marginalizing over the free parameters using Eq. (6.56):

$$p(\boldsymbol{t}|\boldsymbol{x}, \mathcal{D}, \boldsymbol{\alpha}, \boldsymbol{\Sigma}_\beta) = \int p(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{\Sigma}_\beta)\tilde{p}(\boldsymbol{w}|\mathcal{D}, \boldsymbol{\alpha}, \boldsymbol{\Sigma}_\beta)\, d\boldsymbol{w}. \tag{6.57}$$

Due to the nonlinearity of the input-output relation of the neural network in Eq. (6.45), the integration cannot be performed in an analytical way. Approximating the input-output relation by a Taylor-series expansion, where only the constant and the linear terms are considered, gives

$$\tilde{\boldsymbol{y}}(\boldsymbol{x}, \boldsymbol{w}) \approx \boldsymbol{y}(\boldsymbol{x}, \boldsymbol{w}_{MAP}) + \boldsymbol{J}\left(\boldsymbol{w} - \boldsymbol{w}_{MAP}\right), \tag{6.58}$$

where $\boldsymbol{J}$ is the sensitivity of the network output with respect to the free parameters given in Eq. (6.6) evaluated at $\boldsymbol{w} = \boldsymbol{w}_{MAP}$. Substitution of Eq. (6.58) into Eq. (6.45) yields a Gaussian approximation with a mean that is a linear function of the free parameters,

$$p(\boldsymbol{t}|\boldsymbol{x}, \boldsymbol{w}, \boldsymbol{\Sigma}_\beta) \approx \mathcal{N}\left(\boldsymbol{t}|\boldsymbol{y}(\boldsymbol{x}, \boldsymbol{w}_{MAP}) + \boldsymbol{J}\left(\boldsymbol{w} - \boldsymbol{w}_{MAP}\right), \boldsymbol{\Sigma}_\beta\right). \tag{6.59}$$

Using the procedure of completing the square (section C.2), this yields a Gaussian approximation to the predictive distribution

$$p(\boldsymbol{t}|\boldsymbol{x}, \mathcal{D}, \boldsymbol{\alpha}, \boldsymbol{\Sigma}_\beta) \approx \mathcal{N}\left(\boldsymbol{t}|\boldsymbol{y}(\boldsymbol{x}, \boldsymbol{w}_{MAP}), \boldsymbol{\Sigma}(\boldsymbol{x})\right) \tag{6.60}$$

$$\boldsymbol{\Sigma}(\boldsymbol{x}) = \boldsymbol{\Sigma}_\beta + \boldsymbol{J}(\boldsymbol{x})^T \boldsymbol{A}^{-1} \boldsymbol{J}(\boldsymbol{x}). \tag{6.61}$$

6.3.5 Optimization of hyperparameters

So far it has been assumed that the hyperparameters $\boldsymbol{\alpha}$ and the noise covariance $\boldsymbol{\Sigma}_\beta$ are known. In the evidence framework, these hyperparameters are determined in an iterative procedure. The evidence of the hyperparameters is obtained by marginalizing over the weights

$$p(\mathcal{D}|\boldsymbol{\alpha}, \boldsymbol{\Sigma}_\beta) = \int p(\mathcal{D}|\boldsymbol{w}, \boldsymbol{\Sigma}_\beta)\; p(\boldsymbol{w}|\boldsymbol{\alpha})\; d\boldsymbol{w}. \tag{6.62}$$

For simplicity, the integrand $p(\mathcal{D}|\boldsymbol{w}, \boldsymbol{\Sigma}_\beta)\; p(\boldsymbol{w}|\boldsymbol{\alpha})$ is abbreviated with $f(\boldsymbol{w})$. Using Bayes theorem

$$f(\boldsymbol{w}) = p(\boldsymbol{w}|\mathcal{D}, \boldsymbol{\alpha}, \boldsymbol{\Sigma}_\beta)\; p(\mathcal{D}) \tag{6.63}$$

and, with Eq. (6.56), it follows that $\boldsymbol{w}_{MAP}$ is a mode of $p(\boldsymbol{w}|\mathcal{D}, \boldsymbol{\alpha}, \boldsymbol{\Sigma}_\beta)$ and, consequently, of $f(\boldsymbol{w})$ with $\nabla f(\boldsymbol{w})$ equals zero. A Taylor series expansion of $\ln f(\boldsymbol{w})$ with a truncation after the quadratic terms gives:

$$\ln f(\boldsymbol{w}) \approx \ln f(\boldsymbol{w}_{MAP}) - \frac{1}{2}\left(\boldsymbol{w} - \boldsymbol{w}_{MAP}\right)^T A \left(\boldsymbol{w} - \boldsymbol{w}_{MAP}\right), \tag{6.64}$$

where $-\nabla\nabla \ln f(\boldsymbol{w}_{MAP}) = -\nabla\nabla \ln p(\boldsymbol{w}_{MAP}|\mathcal{D}) - \nabla\nabla \ln p(\mathcal{D}) = \boldsymbol{A}$, according to Eqs. (6.54) and (6.63) with $\nabla\nabla \ln p(\mathcal{D}) = 0$, has been used. The Taylor series expansion is performed in the logarithmic scale, since the logarithm of the Gaussian distribution is purely quadratic and the approximation is exact. Taking both sides of Eq. (6.64) as arguments of an exponential function and integrating over $\boldsymbol{w}$ gives:

$$p(\mathcal{D}|\boldsymbol{\alpha}, \boldsymbol{\Sigma}_\beta) = \int f(\boldsymbol{w})\; d\boldsymbol{w} \tag{6.65}$$

$$\approx f(\boldsymbol{w}_{MAP}) \int e^{-\frac{1}{2}\left(\boldsymbol{w} - \boldsymbol{w}_{MAP}\right)^T \boldsymbol{A} \left(\boldsymbol{w} - \boldsymbol{w}_{MAP}\right)}. \tag{6.66}$$

Comparing Eq. (6.66) with the standard form of a Gaussian density function (see section C.1) with its integral over the total domain being equal to one, taking the logarithm and substituting $f(\boldsymbol{w}_{MAP})$, it follows that

$$\ln p(\mathcal{D}|\boldsymbol{\alpha}, \boldsymbol{\Sigma}_\beta) = \ln p(\mathcal{D}|\boldsymbol{w}_{MAP}, \boldsymbol{\Sigma}_\beta) + \ln p(\boldsymbol{w}_{MAP}|\boldsymbol{\alpha}) + \frac{N^w}{2} \ln(2\pi) - \frac{1}{2} \ln |\boldsymbol{A}|, \tag{6.67}$$

with $|\boldsymbol{A}|$ being the determinant of $\boldsymbol{A}$ and $|\boldsymbol{A}|^{-1} = |\boldsymbol{A}^{-1}|$. The joint probability of $p(\mathcal{D}|\boldsymbol{w}_{MAP}, \boldsymbol{\Sigma}_\beta)$ is given using Eq. (6.47)

$$\ln p(\mathcal{D}|\boldsymbol{w}_{MAP}, \boldsymbol{\Sigma}_\beta) = \ln \prod_{m=1}^{M} p(\boldsymbol{t}^{(m)}|\boldsymbol{x}^{(m)}, \boldsymbol{w}_{MAP}, \boldsymbol{\Sigma}_\beta) \tag{6.68}$$

$$= \sum_{m=1}^{M} \ln \left(\frac{1}{(2\pi)^{\frac{N^o}{2}}} \frac{1}{|\boldsymbol{\Sigma}_\beta|^{\frac{1}{2}}} \boldsymbol{e}^{\frac{\Delta^{(m)T} \boldsymbol{\Sigma}_\beta^{-1} \Delta^{(m)}}{2}} \right) \tag{6.69}$$

$$= -\frac{M}{2} N^o \ln(2\pi) - \frac{M}{2} \ln |\boldsymbol{\Sigma}_\beta| - \sum_{m=1}^{M} \frac{\Delta^{(m)T} \boldsymbol{\Sigma}_\beta^{-1} \Delta^{(m)}}{2}, \tag{6.70}$$

with

$$\Delta^{(m)} = \boldsymbol{y}^{(m)} - \boldsymbol{t}^{(m)}. \tag{6.71}$$

Similarly, the logarithm of $p(\boldsymbol{w}_{MAP}|\boldsymbol{\alpha})$ can be expressed using Eq. (6.46) as

$$\ln p(\boldsymbol{w}_{MAP}|\boldsymbol{\alpha}) = -\frac{1}{2} \sum_{j=1}^{N^w} \alpha_j w_j^2 - \frac{N^w}{2} \ln(2\pi) + \frac{1}{2} \sum_{i=1}^{N^w} \ln \alpha_i. \tag{6.72}$$

Substitution of Eqs. (6.70), (6.72) into Eq. (6.67) yields:

$$\ln p(\mathcal{D}|\boldsymbol{\alpha}, \boldsymbol{\Sigma_\beta}) = -\frac{1}{2} \sum_{j=1}^{N^w} \alpha_j w_j^2 - \sum_{m=1}^{M} \frac{\Delta^{(m)T} \boldsymbol{\Sigma}_\beta^{-1} \Delta^{(m)}}{2} - \frac{M N^o \ln(2\pi)}{2} + \frac{M}{2} \ln |\boldsymbol{\Sigma}_\beta^{-1}| + \frac{1}{2} \sum_{i=1}^{N^w} \ln \alpha_i - \frac{\ln |\boldsymbol{A}|}{2}, \tag{6.73}$$

with $-\ln |\boldsymbol{\Sigma}_\beta| = \ln |\boldsymbol{\Sigma}_\beta^{-1}|$. The hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\Sigma}_\beta$ are determined to maximize the probability of the data given the hyperparameters in Eq. (6.73) following a procedure in [238]. Differentiation of Eq. (6.73) with respect to α_i and setting the partial derivative to zero gives:

$$0 = \frac{\partial \ln p(\mathcal{D}|\boldsymbol{\alpha}, \boldsymbol{\Sigma}_\beta)}{\partial \alpha_c} \tag{6.74}$$

$$= -\frac{1}{2} \sum_{j:\alpha_j \epsilon C} w_j^2 + \frac{1}{2} \sum_{j:\alpha_j \epsilon C} \frac{1}{\alpha_j} - \frac{1}{2} \sum_{j:\alpha_j \epsilon C} \boldsymbol{A}_{(jj)}^{-1}), \tag{6.75}$$

where Eq. (C.7) has been used. The Summation is performed over the j, for which the α_j are in the same group C, which means $\alpha_j = \alpha_C \ \forall j : \alpha_j \epsilon C$. Multiplication with α_c and rearranging Eq. (6.75) gives

$$\alpha_c = \frac{\gamma_c}{\sum_{j:\alpha_j \epsilon C} w_j^2} \tag{6.76}$$

$$\gamma_c = k_c - \alpha_c \sum_{j:\alpha_j \epsilon C} \boldsymbol{A}_{(jj)}^{-1}, \tag{6.77}$$

where k_c is the number of elements in group C and γ_c the effective number of parameters in group C. The summation in $\boldsymbol{A}^{-1}_{(jj)}$ is performed over the jth diagonal elements of $\boldsymbol{A}^{-1}$. It is to be noted that Eq. (6.76) is a fixed point iteration, since the right hand side also depends on α_c. In the numerical implementation, this poses no problem, since after an iteration step with an update of the hyperparameters another optimization step for $\boldsymbol{w}_{MAP}$ has to be performed, which changes the weights and due to the nonlinearity of the network response also the covariance matrix $\boldsymbol{A}$.

The update of the noise covariances $\boldsymbol{\Sigma}_\beta$ is obtained in a similar way by differentiation of Eq. (6.73) with respect to $\boldsymbol{\Sigma}_\beta^{-1}$:

$$0 = \frac{\partial \ln p(\mathcal{D}|\boldsymbol{\alpha}, \boldsymbol{\Sigma}_\beta)}{\partial \boldsymbol{\Sigma}_\beta^{-1}} \tag{6.78}$$

$$= \frac{1}{2}\left(\sum_{m=1}^{M} \left(-\Delta^{(m)} \Delta^{(m)^T} \right) + M\boldsymbol{\Sigma}_\beta - \frac{\partial \ln |\boldsymbol{A}|}{\partial \Sigma^{-1}_{\beta(ij)}} \right), \tag{6.79}$$

where the relation

$$\frac{\partial}{\partial \boldsymbol{\Sigma}_\beta^{-1}} \ln |\boldsymbol{\Sigma}_\beta^{-1}| = (\boldsymbol{\Sigma}_\beta)^T \tag{6.80}$$

has been used. The derivative $\dfrac{\partial \ln |\boldsymbol{A}|}{\partial \Sigma^{-1}_{\beta(ij)}}$ is obtained from

$$\frac{\partial \ln |\boldsymbol{A}|}{\partial \Sigma^{-1}_{\beta(ij)}} = \mathrm{tr}\left(\boldsymbol{A}^{-1} \frac{\partial \boldsymbol{A}}{\partial \Sigma^{-1}_{\beta(ij)}} \right). \tag{6.81}$$

Using Eq. (6.55), each entry of the Hessian $\boldsymbol{A}$ can be rewritten as

$$A_{kl} = \frac{1}{2} \frac{\partial^2}{\partial w_k \partial w_l} \left(\sum_{i=1}^{N^w} \alpha_i w_i^2 + \sum_{m=1}^{M} \sum_{i=1}^{N^w} \sum_{j=1}^{N^w} \Delta_i^{(m)} \Delta_j^{(m)} \Sigma^{-1}_{\beta(ij)} \right) \tag{6.82}$$

$$= \delta_{kl}\alpha_k + \frac{1}{2} \frac{\partial}{\partial w_l} \left(\sum_{m=1}^{M} \sum_{i=1}^{N^w} \sum_{j=1}^{N^w} \Sigma^{-1}_{\beta(ij)} \left(\frac{\partial \Delta_i^{(m)}}{\partial w_k} \Delta_j^{(m)} + \Delta_i^{(m)} \frac{\partial \Delta_j^{(m)}}{\partial w_k} \right) \right) \tag{6.83}$$

$$= \delta_{kl}\alpha_k + \frac{1}{2} \left(\sum_{m=1}^{M} \sum_{i=1}^{N^w} \sum_{j=1}^{N^w} \Sigma^{-1}_{\beta(ij)} \left(\frac{\partial^2 \Delta_i^{(m)}}{\partial w_k \partial w_l} \Delta_j^{(m)} + \frac{\partial \Delta_i^{(m)}}{\partial w_k} \frac{\partial \Delta_j^{(m)}}{\partial w_l} + \right. \right.$$
$$\left. \left. \frac{\partial^2 \Delta_j^{(m)}}{\partial w_k \partial w_l} \Delta_i^{(m)} + \frac{\partial \Delta_i^{(m)}}{\partial w_l} \frac{\partial \Delta_j^{(m)}}{\partial w_k} \right) \right). \tag{6.84}$$

with $\Delta^{(m)}$ given in Eq. (6.71). Consequently, the derivative of $\boldsymbol{A}$ with respect to $\Sigma^{-1}_{\beta(ij)}$ can be expressed as

$$\frac{\partial A_{kl}}{\partial \Sigma^{-1}_{\beta(ij)}} = \frac{1}{2} \sum_{m=1}^{M} \left(\frac{\partial^2 y_i(\boldsymbol{x}^{(m)}, \boldsymbol{w})}{\partial w_k \partial w_l} \Delta_j^{(m)} + \frac{\partial y_i(\boldsymbol{x}^{(m)}, \boldsymbol{w})}{\partial w_k} \frac{\partial y_j(\boldsymbol{x}^{(m)}, \boldsymbol{w})}{\partial w_l} + \right. \tag{6.85}$$
$$\left. \frac{\partial^2 y_j(\boldsymbol{x}^{(m)}, \boldsymbol{w})}{\partial w_k \partial w_l} \Delta_i^{(m)} + \frac{\partial y_i(\boldsymbol{x}^{(m)}, \boldsymbol{w})}{\partial w_l} \frac{\partial y_j(\boldsymbol{x}^{(m)}, \boldsymbol{w})}{\partial w_k} \right). \tag{6.86}$$

Using again the approximation of the Hessian matrix, where only the gradient information is used, this can be simplified to

$$\frac{\partial A_{kl}}{\partial \Sigma^{-1}_{\beta(ij)}} \approx \frac{1}{2}\sum_{m=1}^{M}\left(\frac{\partial y_i(\boldsymbol{x}^{(m)},\boldsymbol{w})}{\partial w_k}\frac{\partial y_j(\boldsymbol{x}^{(m)},\boldsymbol{w})}{\partial w_l} + \frac{\partial y_i(\boldsymbol{x}^{(m)},\boldsymbol{w})}{\partial w_l}\frac{\partial y_j(\boldsymbol{x}^{(m)},\boldsymbol{w})}{\partial w_k}\right). \tag{6.87}$$

The sensitivities $\dfrac{\partial y_i}{\partial w_k}$ can be computed using a standard backpropagation scheme. Rearranging Eq. (6.79) gives

$$\boldsymbol{\Sigma}_\beta = \frac{1}{M}\left(\sum_{m=1}^{M}\left(\Delta^{(m)}\Delta^{(m)T}\right) + \frac{\partial \ln|\boldsymbol{A}|}{\partial \boldsymbol{\Sigma}_\beta^{-1}}\right). \tag{6.88}$$

Equation (6.88) is again a fixed point iteration scheme to determine $\boldsymbol{\Sigma}_\beta$ and by inversion $\boldsymbol{\Sigma}_\beta^{-1}$. It is to be noted that the fixed point iteration scheme to determine $bold symbol \Sigma_\beta$ deviates from the standard literature due to the introduction of the full covariance matrix.

6.3.6 Numerical implementation

The proposed method is an iterative strategy for the determination of the hyperparameters $\boldsymbol{\alpha}$, the noise covariance $\boldsymbol{\Sigma}_\beta$ and the weights $\boldsymbol{w}_{MAP}$. The general procedure is summarized in Algorithm 6.3. The noise covariances $\boldsymbol{\Sigma}_\beta$ is initialized with the identity matrix. In order to realize a wide prior distribution of the free parameters, values of 10^{-5} are used for the initialization of α_i. These values correspond to an overfitting in the initial step. In the following iterations, the regularization through the hyperparameters $\boldsymbol{\alpha}$ leads to a regularized final solution. The optimization of the modified objective function in Eq. (6.53) is performed with the preconditioned conjugate gradient algorithm explained in section 6.2.3.3, where the modifications of the gradient and hessian due to the hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\Sigma}_\beta$ in the calculation of the gradient and the hessian are incorporated. The convergence criteria for this optimization step is based on the norm of the gradient in the preconditioned system. Afterwards, an update of the hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is performed and another inner optimization loop of the free parameters $\boldsymbol{w}_{MAP}$ is performed. Convergence in the outer loop is obtained, if the relative modification of the objective function in Eq. (6.53) between two consecutive outer iteration loops is smaller than a certain threshold (10^{-6} is used in the implementation) or the modifications of the hyperparameters are small according to

$$10^{-6} > \sum_{i=1}^{N^\alpha}\begin{cases}\dfrac{(\alpha_i^{(k)}-\alpha_i^{(k-1)})^2}{\alpha_i^{(k-1)^2}} & \alpha_i^{(k-1)} > 1\\ (\alpha_i^{(k)}-\alpha_i^{(k-1)})^2 & \text{otherwise}\end{cases} \tag{6.89}$$

$$10^{-6} > \sum_{i=1}^{N^o}\sum_{j=1}^{N^o}\begin{cases}\dfrac{(\Sigma_{ij}^{(k)}-\Sigma_{ij}^{(k-1)})^2}{\Sigma_{ij}^{(k-1)^2}} & \Sigma_{ij}^{(k-1)} > 1\\ (\Sigma_{ij}^{(k)}-\Sigma_{ij}^{(k-1)})^2 & \text{otherwise.}\end{cases} \tag{6.90}$$

Algorithm 6.3 Bayesian training of neural networks

scale input and outputs of training data to standard normal distributed variables
initialize hyperparameters $\alpha_j = 10^{-5}$ and $\boldsymbol{\Sigma}_\beta = \boldsymbol{I}$, biases $b_i^{(j)} = 0$ and weights $w_{ki}^{(j)}$ according to Eq. (6.44)
repeat
  minimize $-\ln p(\boldsymbol{w}|\mathcal{D}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ in Eq. (6.53) with fixed hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\Sigma}_\beta$
  update hyperparameters $\boldsymbol{\alpha}$ and noise covariance $\boldsymbol{\Sigma}_\beta$ according to Eqs. (6.76), (6.88)
until convergence

It is to be noted that the exact values of the convergence thresholds do not have a significant influence on the final result.

Within the Bayesian determination of the hyperparameters, the number of neurons has only a minor influence. In the numerical implementation, it was advantageous to chose the number of free parameters (weights and biases) smaller than the number of training data, which is given by the number of training samples multiplied by the number of outputs. Consequently, a reasonable upper bound for the number of neurons for a given network architecture (number of inputs/outputs, number of hidden layers) is obtained. In the current work, only one hidden layer has been used. From other tests it was found that a network with one hidden layer performs better for additive input/output relations, whereas a network with two hidden layers performs better for multiplicative input/output relations. The general procedure for training Bayesian neural networks is summarized in Algorithm 6.3.

6.4 Examples of function approximation using neural networks

In order to investigate the quality of the Bayesian approach with respect to the dimensionality of the problem, the generalization quality of the obtained metamodel and the accuracy of the estimates for noise variance and correlation, a test example is set up. It consists of n identical input/output mappings of the form

$$y_i = \sin(x_i) \qquad , x_i \epsilon [-3, 3] \text{ and } i \epsilon \{1, .., n\} . \tag{6.91}$$

Additionally, Gaussian noise is added to the output variables with a prescribed variance and correlation. Unless otherwise specified, the off-diagonal terms of the noise correlation and the prescribed variance of all outputs are identical.

6.4.1 Illustration in 1D

In order to demonstrate certain features of the presented approach, the example is, in a first approach, calculated with a single input and a single output variable. Figure 6.7 illustrates the example with a varying number of training samples and different values for the prescribed noise added to the training samples. Furthermore, the influence of the number of hidden neurons in the single hidden layer is investigated.

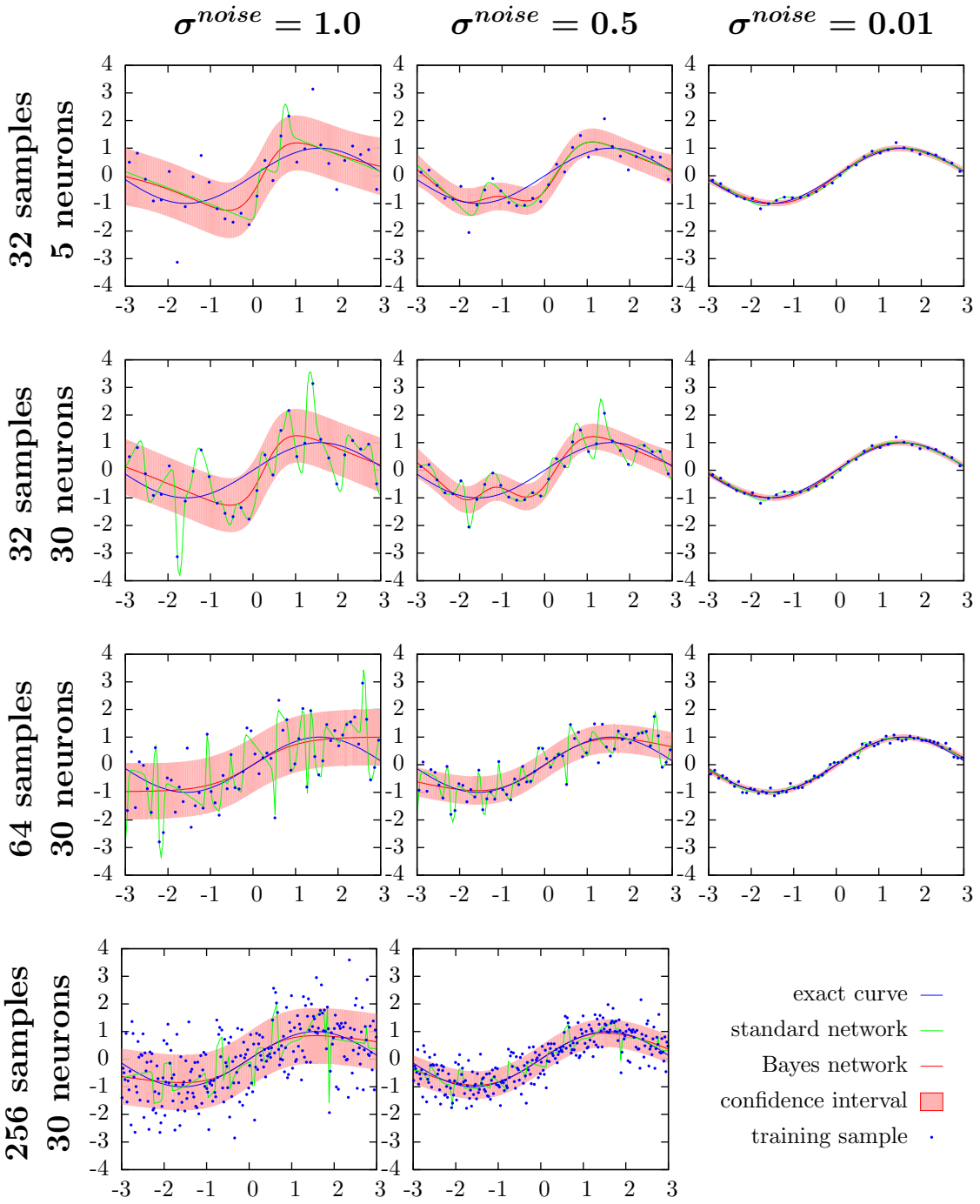


Figure 6.7 Mean and standard deviation of the approximation using a varying number of neurons in the hidden layer and a varying number of training samples disturbed by noise with a prescribed standard deviation.

First of all, it is observed that the number of neurons in the hidden layer does not have a significant influence on the approximation of the output. In contrast to standard neural networks, where a large number of neurons in the hidden layer often leads to overfitting, the results using Bayesian neural networks with 5 and 30 neurons in the hidden layer are almost identical for the example with 32 training samples. It is furthermore observed that the identified noise level of the output can already be accurately determined with a small number of 32 training samples and the identified noise level almost coincides with the prescribed values. With an increased number of training samples, the accuracy of the approximation (which is the mean of the normal distributed response) is increased, while the noise level remains almost constant.

6.4.2 Influence of the dimensionality

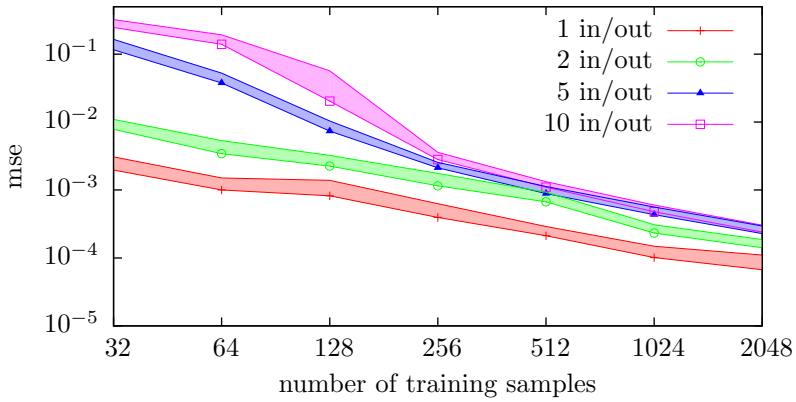
In the first setup of the multidimensional problem, the influence of the dimensionality of the problem (number n of input/output mappings) is investigated. For a prescribed noise variance of 0.1 and independent noise for each output, the dimension of the problem is varied between 1 and 10 with a fixed number of 20 neurons in the single hidden layer. At this point, it is important to realize that the number of neurons in the hidden layer has to be at least as large as the number of important input variables, since otherwise the information of the input is mapped to a lower dimension in the hidden layer. In this example, all input variables are equally important and, consequently, the number of input/output mappings is only increased up to 10. For many practical applications, this poses no restriction, since it is often observed that the number of important input variables (which have a significant influence on the outputs) is much smaller than the actual number of input variables. Furthermore, the balance between the number of training data and the number of free parameters is to be considered. The number of free parameters N^{free} is given by

$$N^{free} = \overbrace{\left(N^i + N^o\right) N^h}^{\text{weights}} + \overbrace{N^h + N^o}^{\text{biases}} + \overbrace{N^i + N^o + 1}^{\boldsymbol{\alpha}} + \begin{cases} \overbrace{\dfrac{N^o(N^o+1)}{2}}^{\boldsymbol{\Sigma}^{-1}} & \text{correlated noise} \\ N^o & \text{uncorrelated noise} \end{cases}, \tag{6.92}$$

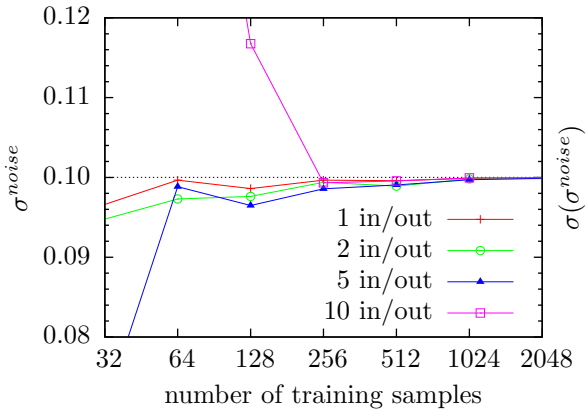
where N^i and N^o are the number of input/output variables and N^h is the number of neurons in the hidden layer. The number of training information is obtained by

$$N^{train} = N^o M,$$

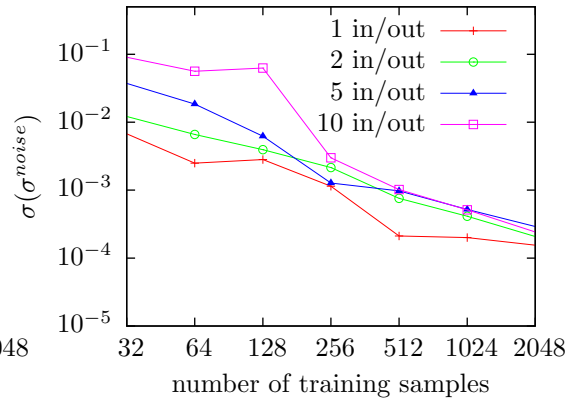
where M is the number of training samples. It is recommended that the number of free parameters is smaller than the number of training information. Due to the regularization in the Bayesian approach, the number of hidden neurons can be increased without strong influence on the results. However, the training of the network becomes computationally more expensive for an increased number of free parameters.



(a) mean square error plus standard deviation for 5 repetitions



(b) Estimated noise level (exact 0.1)



(c) Standard deviation of estimated noise level

Figure 6.8 Influence of number of training samples and dimensionality of the problem on the mean square error and the approximated correlation for uncorrelated variables with a prescribed standard deviation for the added noise of 0.1 and varying dimension of the sin problem in (6.91).

In Fig. 6.8(a), the influence of the number of training samples on the mean square error of an additional test set with 200 samples is plotted, where the test set samples are created in a similar way as the training samples without a noise component. It is realized that the mean square error decreases the more samples are used and supports the theorem that any function can be approximated with a neural network up to a prescribed tolerance. As a consequence this implies that the more training samples are used the better the metamodel is. However, if the training as well as the test data are noisy, the mean square error of the test set will, in the limit, not approach zero, but converge to a value related to the noise in the test data.

It is furthermore observed that the higher the dimension of the input space, the higher is also the approximation error. This seems to be reasonable, since due to an increase of the dimension, the function to be approximated by the metamodel gets much more complex. This can be visualized in a 1D/2D example. Imagine functions $f(x_1, 0)$ and $f(0, x_2)$ to be approximated in 1D with a certain number N of samples $(x_1^{(i)}, x_2^{(i)}), i = 1..N$. For 1D, only two lines along the coordinate axis are to be approximated, whereas for a full 2D metamodel

an approximation over the full space (x_1, x_2) has to be obtained. As a consequence, it can be stated that the higher the dimension of the input space is, the more training samples are required to obtain a model with the same accuracy. The dimension of the output space is not so crucial, since a network with M outputs can be interpreted as M networks with a single output and the influence is only linear. In Fig. 6.8(b), the mean of the estimated noise level over all outputs and 5 repetitions is illustrated, which can be extracted from the diagonal elements of the inverse noise covariance matrix.

$$\sigma_i^{noise} = \sqrt{(\mathbf{\Sigma}_\beta)_{ii}} \tag{6.93}$$

It is realized that already with a small number of 128 samples an accurate estimate of the prescribed noise level 0.1 is obtained. For 10 input/output variables, 256 samples are required. This is due to the fact that the higher the dimension of the input space is, the more difficult it is to distinguish between the real oscillations of a function and noise. Furthermore, the accuracy of the estimate increases by increasing the number of training samples. In Fig. 6.8(c), the standard deviation of the estimated noise level using 5 simulations is plotted. It can be verified that, for this example with 256 samples (for 1-5 variables 128 is already sufficient), the standard deviation of the noise estimate divided by its mean is smaller than 10^{-2}, which allows an accurate determination of the noise level already with a single run (compared to the 5 repetitions performed here to get statistical informations).

6.4.3 Determination of important input and output variables

In practical applications, the engineer is often confronted with a huge set of parameters describing a certain model, without any idea, which of these variables have an influence on a set of certain output variables. The brute force method is often to use all input variables in the metamodel, but this has the disadvantage of increasing the dimensionality of the problem. The Bayesian approach offers the possibility to detect, on the one hand, important input and output variables and, on the other hand, decrease the influence of variables, which are not sensitive to changes of the input parameters. This can be observed by an examination of the hyperparameters $\boldsymbol{\alpha}$. Recall from Eq. (6.46) and Fig. 6.6 that a separate precision parameter α_i is defined for all weights connecting a certain input with the hidden layer and for all weights connecting the hidden layer with a certain output. A large precision parameter implies that weights in this group are all close to zero, since the probability distribution of the weights is given by a normal distribution with zero mean and a standard deviation α_i^{-1}. In the limit of an infinite precision parameter, all weights in this group vanish, which implies that the corresponding input/output parameter can be removed from the network without any changes to the metamodel. Consequently, the precision parameters can be interpreted as a measure of importance for input variables and a measure of the sensitivity of output variables with respect to the input variables.

The applicability of this approach is demonstrated using a modified version of the sin problem in Eq. (6.91):

$$y_i = \frac{1}{i}\sum_{j=1}^{i} \sin(x_j) \qquad , x_j \epsilon [-3, 3] \text{ and } i \epsilon \{1, .., n\}\,. \tag{6.94}$$

It is to be noted that the first input output mapping is identical to Eq. (6.91). The next outputs are given by the average of all previous outputs in Eq. (6.91). Consequently, the first input variable is the most important input variable, since it influences all outputs, whereas the last input variable has only a small influence on the last output.

Within the procedure, the importance of the input variables is detected by the hyperparameter α_i corresponding to each input as illustrated in Fig. 6.9. The first input has the smallest value for the hyperparameter, consequently, the prior distribution of the weights has a larger standard deviation. Vice versa, the last input has the highest hyperparameter and, consequently, the standard deviation of the prior distribution is almost zero, which finally implies that this input has only a small influence on the result. As a result, the Bayesian training procedure allows an automatic extraction of the importance of each input variable. Comparing the results for 10, 20 and 30 neurons in the hidden layer, it is realized that the hyperparameter decreases for an increasing number of neurons. This can simply be attributed to the fact that the input to the transfer function of a neuron in the hidden layer is the weighted sum of input variables and, similar, the input to the transfer function of a neuron in the output layer is the weighted sum of outputs from the hidden layer. If more neurons in the hidden layer are used, the smaller a single weight can be, because a single weight can - from a simplified perspective - be replaced by the sum of two or more identical weights, which are than smaller than the single weight. In fact, the ratio between the hyperparameter α and the number of parameters belonging to this parameter is - for the inputs in the example - approximately independent from the number of neurons in the hidden layer.

As a result, the evaluation of the importance of each input variable from the corresponding hyperparameter α_i should always be interpreted as a relative weighting between the input variables and not as an absolute value, because this value depends on the intrinsic architecture of the network.

The interpretation of the hyperparameters α_i corresponding to the outputs is not so straightforward. On the one hand, the complexity of the approximation for a single output is characterized by its hyperparameter α, since the output is a linear combination of the outputs of the hidden layer. Since the activation function in the hidden layer is a sigmoidal function, its output can be interpreted as a half buckle, which is then weighted and the sum over all neurons in the hidden layer gives the output. As a result, the hyperparameter α_i characterizes the number of buckles required to approximate the function, thus the more complex the output is, the more buckles are required and the smaller the hyperparameter α_i. On the other hand, it is not clear, whether a simple output (simple function without many buckles) is always expressed as a linear combination of a few buckles from the hidden layer. Imagine a network with a single input and two outputs - one output is a linear function whereas the other is a highly nonlinear function (e.g. a sin-function with a period much smaller than the sample space). In order to accurately represent the response of the complex output, the output of the neurons in the hidden layer should capture these buckles. As a consequence, the linear function has to be approximated by a linear combination of the same buckles, which does not necessarily induce the weighting factors to be small, although the complexity of the linear function is much smaller than the sin-function. Finally, this implies that the hyperparameters α_i of the outputs are not only influenced by its corresponding output, but by all other outputs as well which renders the interpretation more difficult.

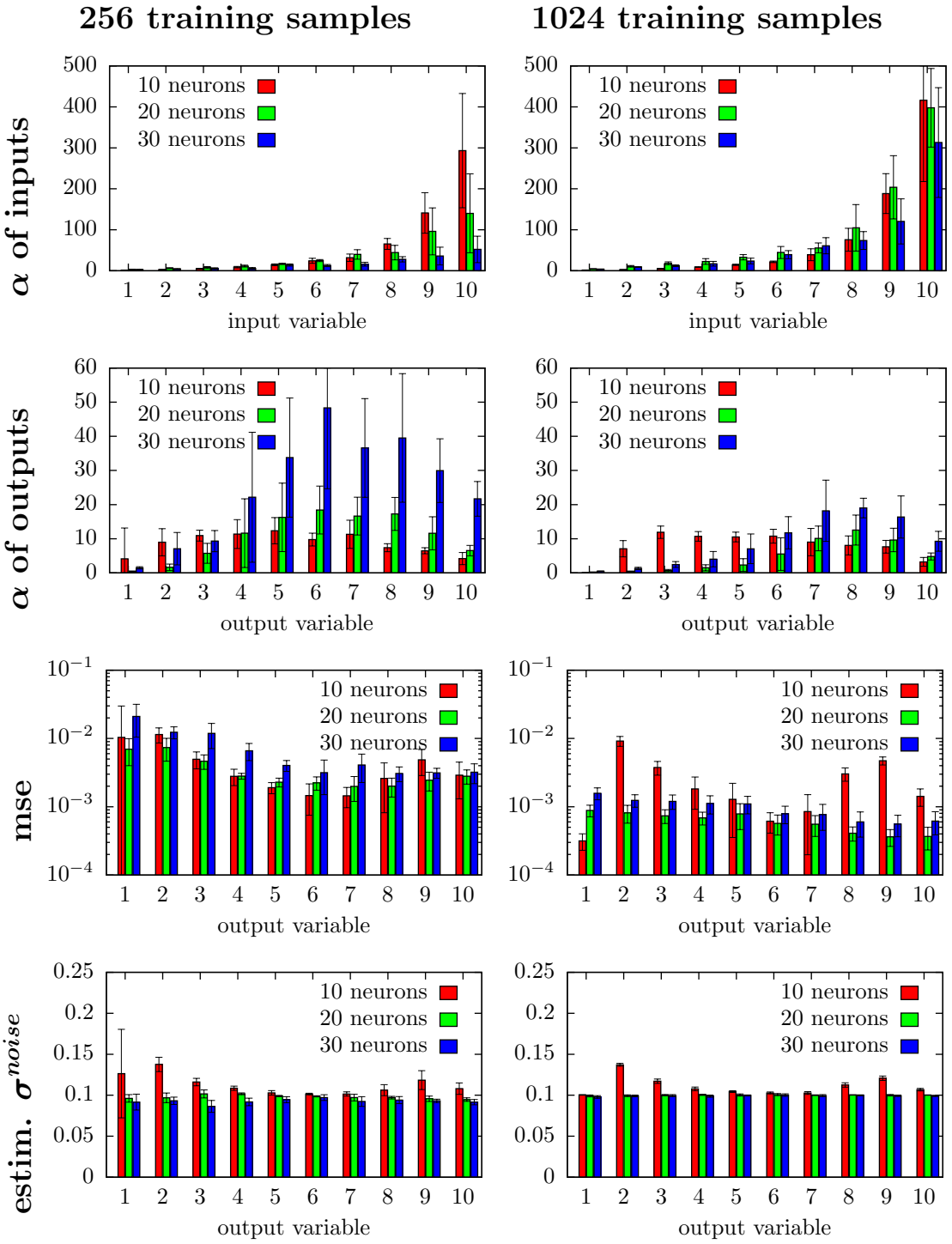


Figure 6.9 Mean and standard deviation for 5 repetitions of the precision parameters α, mean square error and estimated standard deviation of the noise for the modified sin problem in (6.94) with 10 variables and a variable number of neurons in the hidden layer (10,20,30). Independent noise with a standard deviation of 0.1 is used.

Comparing the hyperparameters α_i of the outputs for a varying number of training samples as illustrated in Fig. 6.9, it is realized that for 20 and 30 neurons the hyperparameters decrease with an increasing number of training samples, whereas for 10 neurons almost no difference is obtained. A network with N neurons in the hidden layer maps the input to an intermediate space of dimension N. The output is then a linear combination of functions from this intermediate space. The more neurons are used in the hidden layer, the more complex the function can be. Using only 10 neurons in the hidden layer with 10 inputs as in the example, the complexity of the intermediate space is not sufficient to accurately represent the real function. Using more training samples increases neither the accuracy of the solution nor the complexity of the intermediate space. This can also be realized when comparing the mean square error for 256 and 1024 samples in Fig. 6.9. The accuracy of the solution is not increased when using only 10 neurons in the hidden layer. The situation is different when using 20 or even 30 neurons in the hidden layer. For 256 training samples, the complexity of the approximation is reduced due to the intrinsic regularization procedure, since the number of training data does not allow a complex approximation without the potential problem of overfitting. When the number of training samples is increased to 1024, the confidence in the complex function is higher and the functions are better approximated. Consequently, the number of buckles increases, the weighting factors increase and, as a result, the hyperparameters α_i decrease.

The calculated standard deviation of the noise almost coincides for all tests with the prescribed noise of 0.1. The only exception is the approximation using only 10 neurons in the hidden layer, where the estimated standard deviation of the noise is slightly larger (0.1-0.15), which is still an accurate estimate. The deviation is due to the fact that the estimated covariance matrix, calculated by Eq. (6.88), is directly related to the error in the final approximation of the test data. If there is, due to the not sufficient number of neurons in the hidden layer, a discrepancy between the approximation and the training data, the standard deviation of the noise increases automatically.

6.4.4 Determination of correlated noise in the output parameters

In a final test, the ability of the Bayesian neural network to determine the correlation within the noise of the training samples is investigated. For that purpose, the sin-example from Eq. (6.91) is used with 10 inputs/outputs, 20 neurons in the hidden layer and a prescribed standard deviation of the noise of 0.1, which is added to the training data. All the off-diagonal elements of the linear correlation matrix for the noise were set to the same value, which varies between 0, 0.5 and 0.95. Figure 6.10 illustrates the identified correlation coefficient, where the mean is calculated over all runs (5 runs) and all off-diagonal elements. It is observed that for a small number of training samples the mean of the identified correlation is almost zero, but the standard deviation is large. This is due to the fact that an approximation with 10 inputs/outputs cannot be approximated using only 32 training samples. With a reasonable number of training samples of 128 or even 256 samples, the estimate is very close to the exact value with small scatter. As a consequence, it can be stated that the Bayesian neural network is not only able to evaluate the diagonal elements of the covariance matrix, which characterizes the noise in the training data, but to estimate the full covariance matrix. This is essential in the context of parameter identification as illustrated in the next chapter.

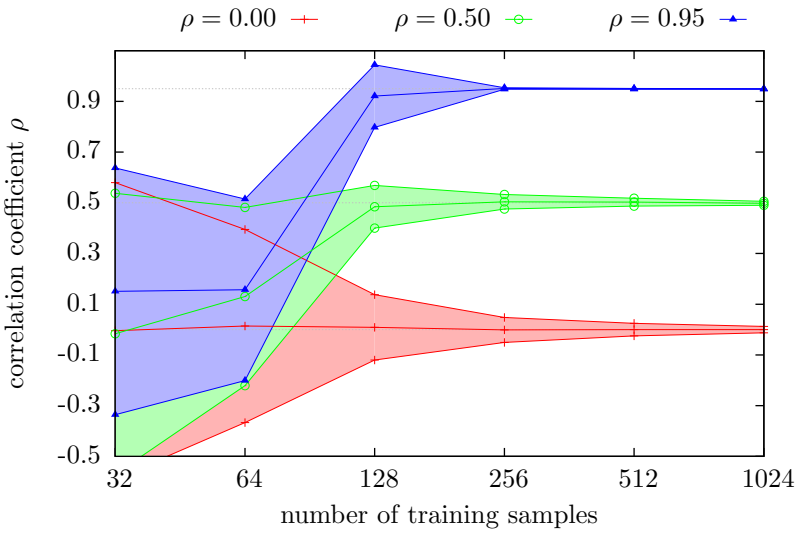


Figure 6.10 Approximated correlation coefficient for the noise with a prescribed standard deviation for the added noise of 0.1 and 10 input/output variables in the sin problem (6.91).

6.5 Support vector machines

Support vector machines are an efficient tool for classification purposes. The origins date back to [239] and a comprehensive introduction can be found in [240]. The principal idea is to define a hyperplane that separates the data points into two classes by further fulfilling the maximum margin principal.

6.5.1 Linear separable case

Support vector machines define a function f

$$f(\boldsymbol{x}) = \boldsymbol{w}\boldsymbol{x} + b, \tag{6.95}$$

where $f(\boldsymbol{x}) = 0$ describes the separating hyperplane between positive and negative data points. Without loss of generality, the training data is given by $(\boldsymbol{x}_i, y_i), y_i \epsilon \{-1, 1\}, \boldsymbol{x}_i \epsilon \mathbb{R}^d, i = 1..L$, where L corresponds to the number of training samples. Assume that there exist a separating hyperplane as illustrated in Fig. 6.11, which separates the positive from the negative data points, so that:

$$\boldsymbol{x}_i\boldsymbol{w} + b \geq +1 \quad \text{for} \quad y_i = +1 \tag{6.96}$$

$$\boldsymbol{x}_i\boldsymbol{w} + b \leq -1 \quad \text{for} \quad y_i = -1, \tag{6.97}$$

which can be combined into a single inequality

$$y_i \left(\boldsymbol{x}_i\boldsymbol{w} + b\right) - 1 \geq 0 \quad \forall i = 1..L. \tag{6.98}$$

The shortest distance from the separating hyperplane to a positive (negative) point is denoted by $d^+(d^-)$. Define the margin as $d^+ + d^-$. In general, there exists an infinite number of

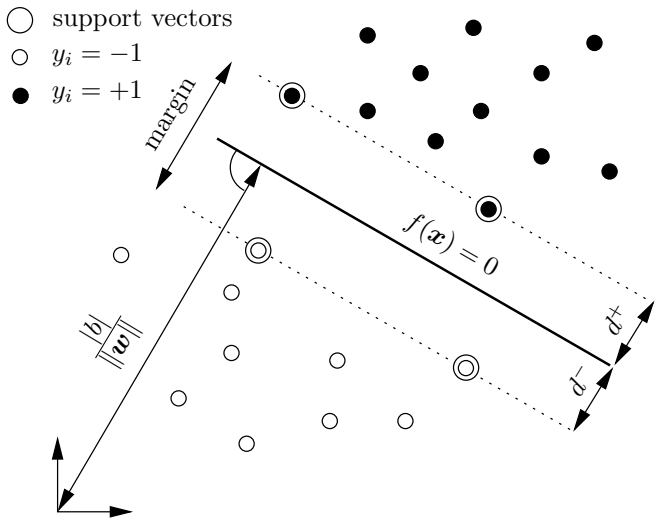


Figure 6.11 Separating hyperplane for linear separable problem.

separating hyperplanes. In the approach for the support vector machines, the hyperplane which maximizes the margin is chosen. The separating hyperplane is given by $\boldsymbol{w}\boldsymbol{x} + b = 0$ and $|b|/\|\boldsymbol{w}\|$ is the perpendicular distance to the origin. In a similar way, the hyperplane for the points which fulfill Eqs. (6.96) and (6.97) exactly can be evaluated and the distances to the origin are given by

$$\boldsymbol{w}\boldsymbol{x} + b = \pm 1 \tag{6.99}$$

$$d_{o\pm} = \frac{|\pm 1 - b|}{\|\boldsymbol{w}\|}. \tag{6.100}$$

As a result, $d^+ = d^- = 1/\|\boldsymbol{w}\|$ and the margin is $2/\|\boldsymbol{w}\|$. Consequently, the following equivalent optimization problem has to be solved:

$$\min \frac{1}{2}\|\boldsymbol{w}\|^2 \tag{6.101}$$

$$\text{subject to } y_i\left(\boldsymbol{x}_i\boldsymbol{w} + b\right) - 1 \geq 0 \quad \forall i = 1..L, \tag{6.102}$$

where the factor $1/2$ is added to obtain the standard form of a quadratic programming minimization problem. The Lagrange formulation for this problem is given by

$$L_p = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{N} \alpha_i \left(y_i(\boldsymbol{x}_i\boldsymbol{w} + b) - 1\right) \tag{6.103}$$

$$\text{with } \alpha_i \geq 0 \quad \forall i = 1..L. \tag{6.104}$$

This is a convex quadratic programming problem with the Lagrange multipliers α_i. Equivalently, the dual problem can be solved (section C.4.2):

$$\max \; -\frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L} \alpha_i\alpha_j y_i y_j \boldsymbol{x}_i\boldsymbol{x}_j + \sum_{i=1}^{L} \alpha_i \tag{6.105}$$

$$\text{subject to } \alpha_i \geq 0 \quad \forall i = 1..L \tag{6.106}$$

$$\boldsymbol{w} = \sum_{i=1}^{L} \alpha_i y_i \boldsymbol{x}_i \tag{6.107}$$

$$0 = \sum_{i=1}^{L} \alpha_i y_i. \tag{6.108}$$

There exists a Lagrange multiplier for each training point. The training points for which the corresponding Lagrange multiplier in the optimal solution are nonzero are called the support vectors, which are located on the hyperplane $\boldsymbol{x}\boldsymbol{w} + b = \pm 1$. Having solved the dual problem, $\boldsymbol{w}$ can directly be determined from Eq. (6.107). The bias parameter b is determined from the first order Karush-Kuhn-Tucker (KKT) conditions of the primal problem

$$\alpha_i(y_i(\boldsymbol{w}\boldsymbol{x}_i + b) - 1) = 0 \quad \forall i. \tag{6.109}$$

Theoretically, any of the equations in (6.109) can be used to determine b although, due to numerical reasons, the mean value from all these equations is a better choice.

6.5.2 Linear nonseparable case

In the general case, the training points do not necessarily define a linearly separable problem. Consequently, non negative slack variables are introduced [241] to relax the constraints given in Eqs. (6.96), (6.97)

$$\boldsymbol{x}_i\boldsymbol{w} + b \geq +1 - \xi \text{ for } y_i = +1 \tag{6.110}$$

$$\boldsymbol{x}_i\boldsymbol{w} + b \leq -1 + \xi \text{ for } y_i = -1. \tag{6.111}$$

By adding the term $C \sum \xi_i$ to the objective function, the slack variables are considered in the minimization procedure. For sufficiently large C, the number of training errors is minimized and the separation of the remaining data with maximum margin is obtained. The parameter C is a userdefined constant and 10^6 is used in the implementation.

This choice of the objective function has the advantage that neither the slack variables nor their corresponding Lagrange multipliers appear in the dual problem as shown in appendix section C.4.3. The dual problem is given by

$$\max \; -\frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i \boldsymbol{x}_j + \sum_{i=1}^{L} \alpha_i \tag{6.112}$$

$$\text{subject to } 0 \leq \alpha_i \leq C \quad \forall i = 1..L \tag{6.113}$$

$$\boldsymbol{w} = \sum_{i=1}^{L} \alpha_i y_i \boldsymbol{x}_i \tag{6.114}$$

$$0 = \sum_{i=1}^{L} \alpha_i y_i \tag{6.115}$$

$$-C = -\alpha_i - \beta_i. \tag{6.116}$$

The only difference compared to the separable problem is the additional upper bound of the Lagrange multiplier α_i. Similar to the previous derivation, the first order KKT conditions of the primal problem can be used to determine the bias parameter b:

$$\alpha_i \left[y_i(\boldsymbol{x}_i \boldsymbol{w} + b) - 1 + \xi_i \right] = 0 \tag{6.117}$$

$$\beta_i \xi_i = 0. \tag{6.118}$$

Using Eqs. (6.116) and (6.118), it follows that $\xi = 0$ for $0 \leq \alpha_i < C$. Consequently, any training point with $0 \leq \alpha_i < C$ can be used to determine b with Eq. (6.117) and $\xi_i = 0$, but from a numerical point of view it is advantageous to use the mean of all these training points.

6.5.3 Nonlinear case

In the previous section, a derivation for the linear case was given. However, many problems are not linearly separable. In order to generalize the procedure to the nonlinear case, a transformation Φ of the data to an intermediate space $\mathbb{H}$ is introduced.

$$\Phi : \mathbb{R}^d \rightarrow \mathbb{H} \tag{6.119}$$

Since the dual optimization problem in Eq. (6.112) depends only on dot products between the training points, an equivalent formulation in the intermediate space gives:

$$\max \; -\frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} \alpha_i \alpha_j y_i y_j \phi(\boldsymbol{x}_i) \phi(\boldsymbol{x}_j) + \sum_{i=1}^{L} \alpha_i \tag{6.120}$$

$$\text{subject to } 0 \leq \alpha_i \leq C \quad \forall i = 1..L \tag{6.121}$$

$$\boldsymbol{w} = \sum_{i=1}^{L} \alpha_i y_i \phi(\boldsymbol{x}_i) \tag{6.122}$$

$$0 = \sum_{i=1}^{L} \alpha_i y_i \tag{6.123}$$

$$-C = -\alpha_i - \beta_i. \tag{6.124}$$

Assume that there is a kernel function K such that $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_j)$. Since the objective function only depends on dot products $\phi(\boldsymbol{x}_i)\phi(\boldsymbol{x}_j)$, which can be replaced by the kernel function K, the transformation itself does not have to be performed. A common kernel function is

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2R}}, \tag{6.125}$$

where R is a userdefined influence radius. The corresponding intermediate space $\mathbb{H}$ is infinite dimensional as is the weight vector $\boldsymbol{w}$.

In order to determine an approximate function value for a test point $\boldsymbol{s}$, Eq. (6.95) is evaluated in the intermediate space $\mathbb{H}$. By substitution of Eq. (C.19), the following formula is obtained:

$$f(\boldsymbol{s}) = \boldsymbol{w}\phi(\boldsymbol{s}) + b = \sum_{i=1}^{L} \alpha_i y_i \phi(\boldsymbol{x}_i)\phi(\boldsymbol{s}) + b \tag{6.126}$$

$$= \sum_{i=1}^{L} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{s}) + b \tag{6.127}$$

Again, the explicit calculation of $\boldsymbol{w}$ and the transformation Φ is not required. Similar to the linear non-separable case, the bias parameter b is determined from the first order KKT conditions, which are given by

$$0 = \alpha_i \left[y_i \left(\phi(\boldsymbol{x}_i)\boldsymbol{w} + b \right) - 1 + \xi_i \right] = \alpha_i \left[y_i \left(\phi(\boldsymbol{x}_i) \sum_{j=1}^{L} \alpha_j y_j \phi(\boldsymbol{x}_j) + b \right) - 1 + \xi_i \right] \tag{6.128}$$

$$= \alpha_i \left[y_i \left(\sum_{j=1}^{L} \alpha_j y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) + b \right) - 1 + \xi_i \right] \tag{6.129}$$

$$0 = \beta_i \xi_i. \tag{6.130}$$

by additionally substituting $\boldsymbol{w}$ using Eq. (6.122). Only those training points in Eq. (6.129) are averaged to calculate b, for which $0 \leq \alpha_i < C$ and, consequently $\xi_i = 0$.

6.5.4 Training

For a given set of data points, the optimization problem in Eqs. (6.120)-(6.123) has to be solved. In the implementation used for this work, this problem is solved using sequential minimal optimization [242]. The basic idea is to divide the full optimization problem into a set of simpler optimization problems. The simplest problem includes solving for two Lagrange multipliers α_1 and α_2, since the Lagrange multipliers must obey the linear relation in Eq. (6.123). It is to be noted that α_1 and α_2 are chosen with an heuristic approach from the set of all Lagrange multiplies. The bounds of the subproblem are visualized in Fig. 6.12. The idea is to express the Lagrange multiplier α_1 as a function of α_2 using the linear constraint relation. Consequently, the bounds on α_2 can be expressed as

$$\alpha_2^{(t+1)} \epsilon \left[L, H \right] = \begin{cases} \left[\max(0, \alpha_2^{(t)} - \alpha_1^{(t)}), \min(C, C + \alpha_2^{(t)} - \alpha_1^{(t)}) \right] & y_1 \neq y_2 \\ \left[\max(0, \alpha_2^{(t)} + \alpha_1^{(t)} - C), \min(C, \alpha_2^{(t)} + \alpha_1^{(t)}) \right] & y_1 = y_2 \end{cases}. \tag{6.131}$$

Furthermore, the second derivative η of the objective function along the linear constraint direction with respect to α_2 can be calculated:

$$\eta = \frac{do^2}{d\alpha_1^2} = \left(\frac{\partial}{\partial \alpha_2} + \frac{\partial \alpha_1}{\partial \alpha_2} \frac{\partial}{\partial \alpha_1} \right) \left(\frac{\partial o}{\partial \alpha_2} + \frac{\partial \alpha_1}{\partial \alpha_2} \frac{\partial o}{\partial \alpha_1} \right) \tag{6.132}$$

$$= K(\boldsymbol{x}_1, \boldsymbol{x}_1) + K(\boldsymbol{x}_2, \boldsymbol{x}_2) - 2K(\boldsymbol{x}_1, \boldsymbol{x}_2), \tag{6.133}$$

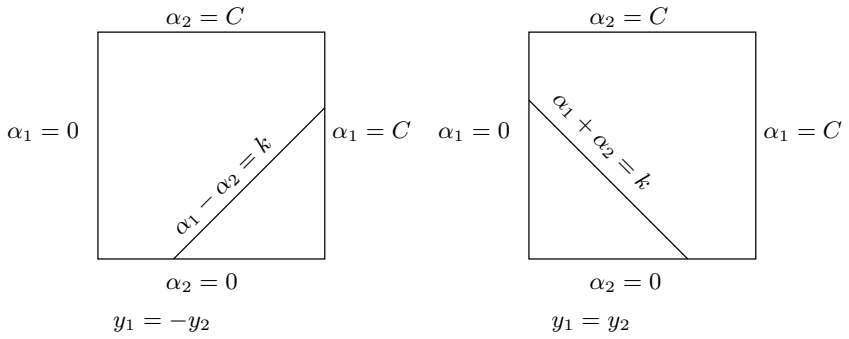


Figure 6.12 Bounds of the subproblem in sequential minimal optimization with two Lagrange multipliers.

where o is the objective function from Eq. (6.120). It is to be noted that the objective function is, under normal circumstances, positive definite. Otherwise, the problem can also be solved as explained in detail in [242]. The gradient along the search direction can be calculated from Eq. (6.120) using Eq. (6.126) and, finally, the solution of the unconstrained quadratic problem is obtained by

$$\alpha_{2,un}^{(t+1)} = \alpha_2^{(t)} + \frac{y_2(E_1 - E_2)}{\eta} \tag{6.134}$$

where $E_i = f(\boldsymbol{x}_i) - y_i$ is the error for the i-th training sample. Clipping α_2 to the bounds and substition for α_1 finally results in:

$$\alpha_2^{(t+1)} = \begin{cases} H & \text{if } \alpha_{2,un}^{(t+1)} \geq H \\ \alpha_{2,un}^{(t+1)} & \text{if } L < \alpha_2^{(t+1)} < H \\ L & \text{if } \alpha_{2,un}^{(t+1)} \leq L \end{cases} \tag{6.135}$$

$$\alpha_1^{(t+1)} = \alpha_1^{(t)} + y_1 y_2 \left(\alpha_2^{(t)} - \alpha_2^{(t+1)} \right). \tag{6.136}$$

For further details on the heuristics and details of the implementation see [242].

Chapter 7

Parameter identification

7.1 Introduction

Complex models are often used to accurately simulate real life problems. This requires the identification of the model parameters, which are often not physically interpretable and, consequently, a direct evaluation from experimental data is not straightforward. In general, these problems are called inverse problems to the direct problem, in which, for a given parameter set, the output of the numerical model is computed. These inverse problems are often ill-posed, which can be due to many different reasons. First of all, the number of data sets might not be sufficient to determine all parameters or parameters are to be determined that do not have an influence on the measured output values. A second problem are measurement errors due to noisy data and, last but not least, the problem might be intrinsically ill-posed, e.g. if several parameter configurations lead to the same set of outputs. In standard parameter identification procedures, a certain error measure between the numerical and the experimental data set is defined. This error is then minimized by modification of the material parameters in the numerical simulation using different models as e.g. gradient based methods or genetic algorithms. Finally, a set of parameters for the numerical model is obtained that seems to best simulate the experimental data, which might be e.g. load-displacement curves, eigenfrequencies or mode shapes. A general overview of different methods for parameter identification is given in [243, 244, 245, 246]. Frederiksen determined unknown material parameters of a numerical model using a least squares approach [247], whereas Soares solved a similar problem using an inverse parameter identification procedure based on neural networks [248]. Araujo determines the material parameters of composite material from dynamic test [249], [250, 251, 252] identified material parameters of a viscoplastic model using complex test with a non-uniform stress state or cyclic loading and Lefik identified parameters for an elasto-plastic model of a superconducting cable under cyclic loading [253]. Another field of application are monitoring problems [254, 255, 256, 257, 258, 259], where often based on modal data changes with respect to a reference state should be identified.

In general, most parameter identification procedures have the severe disadvantage that no information about the accuracy of the estimate is given. If a certain parameter has no influence on the given test data, or two parameters influence the response in the same way, the experiment should be modified in order to determine these parameters separately. A simple example is a material with different tensile and compressive strength. Obviously, an experimental tensile test will only allow the identification of the tensile strength and an additional compression test is required. For complex material models, the interconnection between material parameter and corresponding tests required to identify that parameter is

Algorithm 7.1 Design of experiments using Bayesian neural networks

repeat
  Design the experiments (specimen geometry, loading, boundary conditions).
  Build up parameterized numerical model and define admissible parameter space.
  Perform latin hypercube sampling to create sets of material parameters
    using the assumption of uncorrelated, uniformly distributed variables.
  Calculate the response for all sets of parameters using the numerical model.
  Train the Bayesian neural network using the responses
    as inputs and the material parameters as output parameters.
  Evaluate the uncertainty of the parameter estimates and their correlation.
until Quality of the parameter estimate is sufficient.
Determine the material parameters using the results from the real experiments
  as input to the Bayesian neural network.

not straightforward, and often a trial and error procedure enhanced by engineering knowledge is performed. In this context, a new methodology is proposed that offers the possibility to estimate the parameter and the precision of the estimate. As a result, the design of a set of experiments, which are sufficient to estimate all parameters of the model, is possible a priori, i.e. having only the numerical model without any experimental data.

At the beginning, the general procedure for the parameter identification method using Bayesian neural networks is explained. Afterwards, the algorithm is applied to two examples - a simple material model in 1D with 3 parameters (Young's modulus, tensile strength and fracture energy) to demonstrate the influence of certain parameters in the model and, finally, a mesoscale model of concrete with 6 material parameters.

7.2 General procedure

The general procedure of the proposed parameter identification method using neural networks can be decomposed into several steps - generation of training data by sampling the material parameters, computation of the response using these parameters in the numerical model, Bayesian training of the neural network using the numerical response as input parameters and the material parameters as outputs and, finally, the interpretation of the trained neural network model. The method can additionally be used to design a set of experiments that has to be carried out in order to determine all parameters of a numerical model. The advantage of the proposed procedure is that it can be carried out a priori having only the numerical model, i.e. without any experimental data sets. The algorithm for the design of experiments is summarized in Algorithm 7.1.

7.2.1 Generation of training data for the Bayesian neural network

At first, a parameterized numerical model has to be build up. From engineering knowledge, the parameter space (i.e. min and max values) for these variables has to be defined. Afterwards, a latin hypercube sampling as discussed in section 5.2.3.2 is applied to sample the

parameters with a uniform distribution in the prescribed range without any correlation. It is to be noted that the procedure works also for correlated, non-uniform distributions, but due to the lack of prior knowledge the proposed joint probability density function is advantageous. Furthermore, this distribution ensures that the same density of support points is achieved in the space of admissible parameter configurations. For each set of material parameters, the response of the numerical model has to be calculated, e.g. displacements, reaction forces or full load-displacement curves, eigenfrequencies, modeshapes etc. These output parameters serve as inputs to the neural network, whereas the corresponding set of material parameters is used as the output of the network. The number of latin hypercube samples has to be determined a priori. A guideline for the choice might be the example of the sin function in section 6.4, where the accuracy of the estimates is plotted against the number of samples and the number of input/outputs.

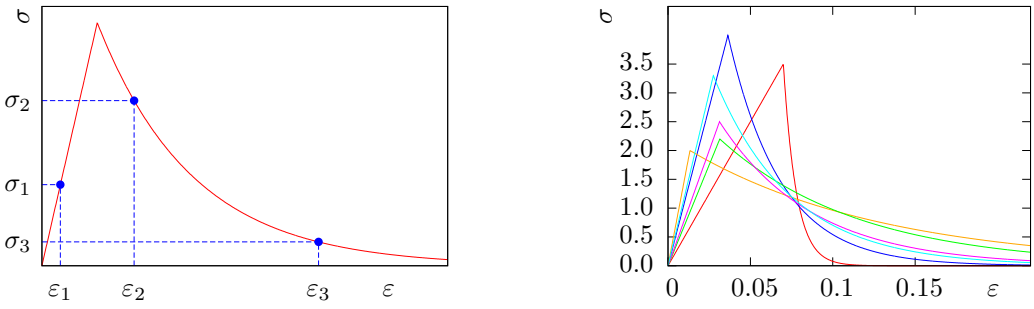
A second question is the design of the network. In all parameter identification examples in this work, a network with a single hidden layer has been used, since it can be shown that any function on a compact support can be approximated with a network with a single hidden layer up to any prescribed tolerance [260]. The maximum number of neurons in the hidden layer can be approximated by the condition that the number of free parameters in the network is approximately equal to the number of training data (number of training samples multiplied by the number of outputs). Although the Bayesian network naturally performs a regularization, the computational effort is much larger for large networks. Finally, the network is trained using the Bayesian procedure described in section 6.3. It is to be noted that a scaling procedure is applied a priori to the training data in order to have standard normal distributed input/output variables.

7.2.2 Interpretation of the results from the neural network

One of the key features of Bayesian neural networks is the ability to estimate the noise level in the data and the quality of the approximation. The quality of the approximation for an input vector $\boldsymbol{x}$ can be calculated from Eq. (6.60). It is to be noted that the covariance matrix $\boldsymbol{\Sigma}$ for the input depends on two summands, the constant covariance of the noise $\boldsymbol{\Sigma}_\beta$ and the part depending on the Hessian of the posterior distribution $\boldsymbol{A}$ and the Jacobian $\boldsymbol{J}(\boldsymbol{x})$. The quality of the approximation is determined by this covariance matrix, from which the correlation coefficients $\boldsymbol{\rho}$ using Eq. (5.11) and the standard deviations σ_i of each output (the material parameters to be estimated) can be derived:

$$\sigma_i = \sqrt{(\boldsymbol{\Sigma}_\beta)_{ii}}. \tag{7.1}$$

If the Bayesian network is used for the design of experiments, the question is, how accurate can each parameter be determined. The dominating influence has the noise covariance $\boldsymbol{\Sigma}_\beta$ so that, in a first approximation, the standard deviation and the correlation coefficient can be calculated using only $\boldsymbol{\Sigma}_\beta$. For a more detailed analysis, the covariance matrix can be averaged over all training samples. It is to be noted that the covariance matrix is calculated in the normalized space, thus rendering all the parameter variances comparable.



(a) Training samples (b) Latin hypercube samples

Figure 7.1 Training samples of the softening material in tension.

7.3 Parameter identification of an artifical softening material in tension

In order to illustrate the method and estimate the influence of certain parameters, a simple example as illustrated in Fig. 7.1 is used. It can be interpreted as a tension test of a cohesive element with a pre-peak elastic stiffness E, a tensile strength f_{ct} and a fracture energy G, which corresponds to the area under the curve:

$$\sigma = \begin{cases} E\varepsilon & \varepsilon < \varepsilon_0 \\ f_{ct} e^{\frac{\varepsilon_0 - \varepsilon}{\varepsilon_f - \varepsilon_0}} & \text{otherwise} \end{cases} \tag{7.2}$$

$$\varepsilon_0 = \frac{f_{ct}}{E} \tag{7.3}$$

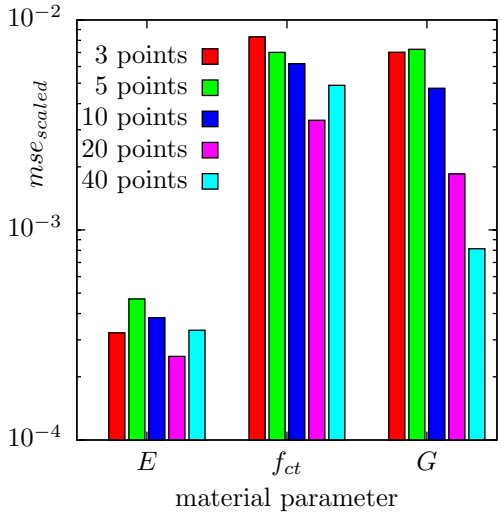
$$\varepsilon_f = \frac{G}{f_{ct}} + \frac{f_{ct}}{2E}. \tag{7.4}$$

It is to be noted that $\varepsilon_f > 0$, otherwise the elastic energy is larger than the total energy, which is not possible. In Fig. 7.1(a), a load-displacement curve for a single parameter configuration is plotted. For prescribed values ε_i, the corresponding values σ_i are used as inputs, whereas the set of material parameters is used as outputs of a training sample. Performing a latin hypercube sampling with a prescribed number of samples, a set of training samples is obtained as illustrated in Fig. 7.1(b).

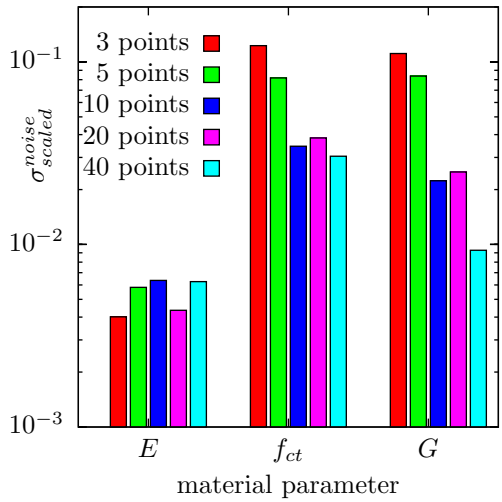
The intervals used for the sampling of the parameters are $I_E = [50, 150]$, $I_{f_{ct}} = [2, 4]$ and $I_G = [2, 4]$. Different setups, where the number of samples, the number of strains and thus the number of input points and the position of the strains is varied, are investigated.

7.3.1 Influence of the number of points in the load-displacement curve

In a first test, the influence of the number of input points (strains ε_i) on the parameter estimation is investigated. A network with 50 neurons in the hidden layer is trained using 100 samples. The number of strains is varied from 3 to 40, where the positions are equally



(a) Scaled mean square error of test set



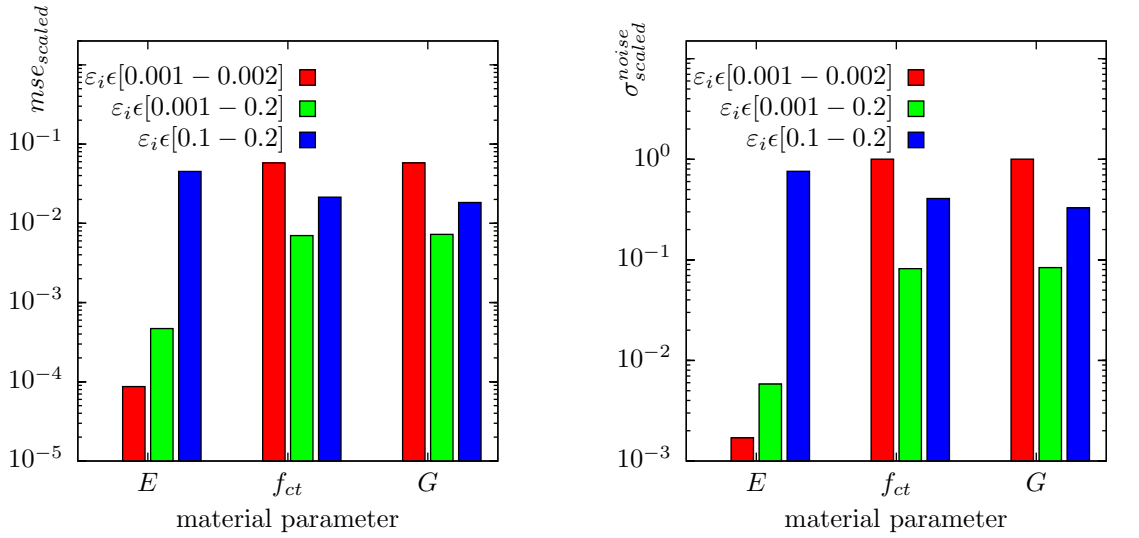
(b) Mean of noise variance for all training data in the normalized space

Figure 7.2 Influence of the number of input variables equally spaced in the range 0.001-0.2 using 100 samples and 50 neurons in the hiddenlayer.

spaced in the interval [0.001,0.2]. The left border 0.001 has been chosen in order to assure that at least one input point is in the linear elastic region. An additional test of 100 samples with the same distribution of the material parameters as the training set has been used to calculate the mean square error in Fig. 7.2(a). It is observed that the Young's modulus can be much better approximated than the tensile strength and the fracture energy. In general, it can be said that the more input points are used, the better the approximation is. This result might be surprising, because by increasing the dimension of the input space the number of free parameters and thus the complexity of the network increases. This is due to the fact that due to the additional input points additional information about the function to be approximated is obtained, which leads to a better parameter estimate, although as detailed in section 7.5, the numerical model must be able to represent the experimental curves. The decrease of the accuracy from 3 to 5 points and from 10 to 20 points is due to the specific problem. When using only 3 input points, there is only one point in the elastic domain, whereas the second and third point are always in the post-peak region. An increase to 5 points decreases the distance between these points and the second point might be located in the pre-peak or the post-peak region. Consequently, the second input point is influenced not only by the Young's modulus, but also by the other material parameters, which adds a correlation between these parameters (Assume that only the second point is present as a single input, then there are multiple parameter sets that give the same output). In general, it is advantageous to decouple the material parameters, i.e. that one input point influences only a single material parameter.

7.3.2 Influence of position of points in the load-displacement curve

This problem is further analyzed in a second setup, where the position of 5 input points is varied. The results are illustrated in Fig. 7.3. If only strains in the post-peak region ($\varepsilon_i \epsilon [0.1, 0.2]$)

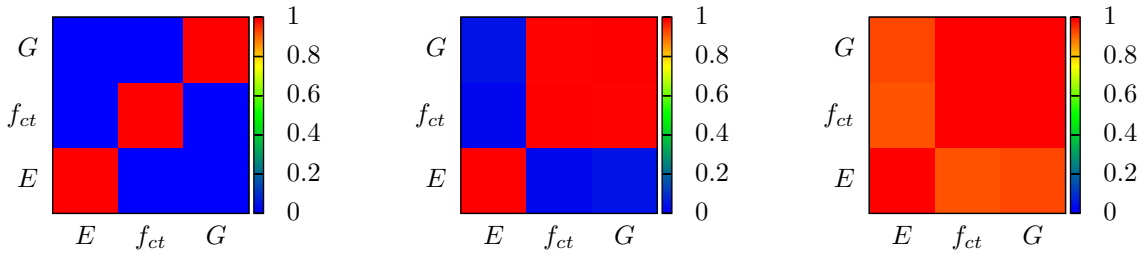


(a) Scaled mean square error of test set

(b) Mean of noise variance for all training data scaled with the variance of the material parameters

Figure 7.3 Influence of the position of 5 input points in the load-displacement curve using 100 samples and 50 neurons in the hidden layer.

are used, the parameters cannot be identified (the variance of the parameter estimates approximately equals the variance of the training set), which corresponds to $\sigma^{noise}_{scaled} \approx 1$. If only strains in the pre-peak region ($\varepsilon_i\epsilon[0.001, 0.002]$) are used, the estimate for the Young's modulus is almost exact, but all the other parameters cannot be determined, since they do not have an influence on the load-displacement curve. This is also realized when looking at the correlation coefficients of the material parameter estimates illustrated in Fig. 7.4. In the case of input points only in the post-peak region, all parameters are correlated. This result is not astonishing, because all parameters influence the points in the post-peak region. When using points in the post- as well as in the pre-peak region, only the fracture energy and the tensile strength are correlated, since the Young's modulus can be solely determined from a point in the pre-peak region, which does not have an influence on the other parameters. In the

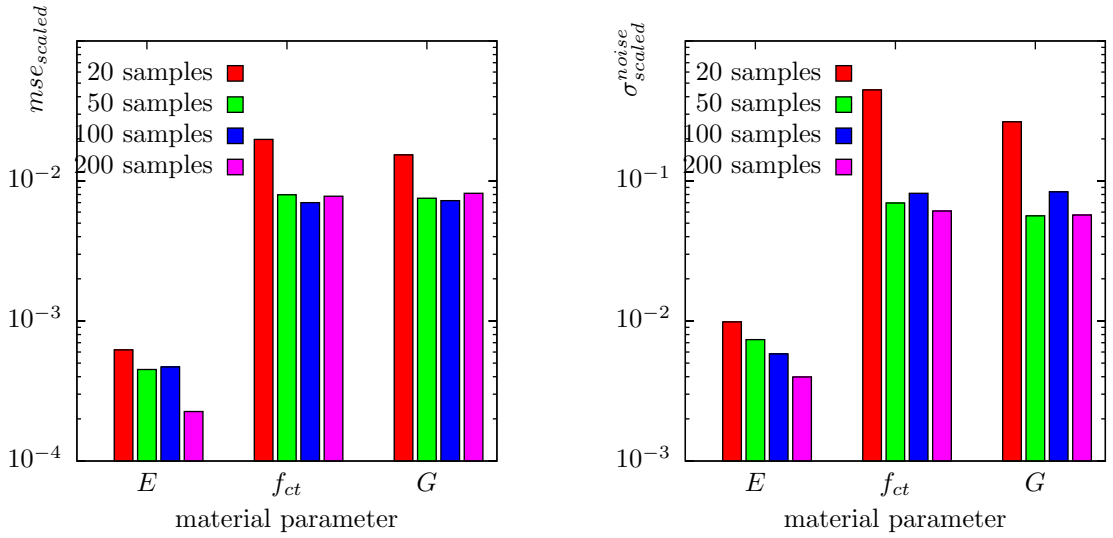


(a) Samples in [0.001,0.002]

(b) Samples in [0.001,0.2]

(c) Samples in [0.1,0.2]

Figure 7.4 Influence of the position of 5 input points on the correlation of the noise corresponding to the identified output parameters using 100 samples and 50 neurons in the hiddenlayer.



(a) Scaled mean square error of test set

(b) Mean of noise variance for all training data scaled with the variance of the material parameters

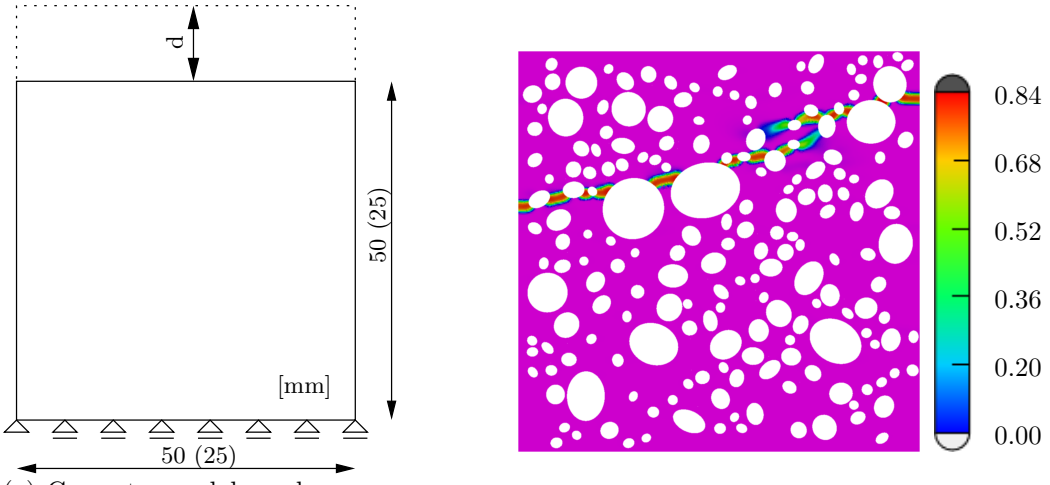
Figure 7.5 Influence of the number of training samples using 5 input points in the load-displacement curve equally spaced in the range 0.001-0.2 with 50 neurons in the hidden layer.

third case of points only in the pre-peak region, the parameters are all uncorrelated, but this information is not of any additional value, since the fracture energy and the tensile strength cannot be determined at all.

For this simple problem, these results are not astonishing. However, for a complex material and a complex test with non-uniform stress states, it might not be straightforward to decide, which points on the load-displacement curve influence a certain parameter. Furthermore, the method is not limited to a single experiment, but there might also be a set of experiments with e.g. $\varepsilon_{1..3}$ coming from the first experiment and $\varepsilon_{4..6}$ from a second etc. As a result, this automatic procedure allows to determine, which parameters can be determined by a specific set of tests. The influence of adding or removing a single test can also be used to reduce the number of required tests to a minimum, thus decreasing the cost of the experiments.

7.3.3 Number of training samples

In the last setup, the influence of the number of samples is investigated, which is illustrated in Fig. 7.5. Obviously, the estimate gets more accurate the more samples are used. In particular, using only 20 samples are not sufficient for an accurate parameter estimation. However, increasing the number of samples from 50 to 200 does not have a significant influence on the accuracy of the material parameter estimates, and it can be concluded that in this case with only 3 parameters to estimate 50-100 samples are sufficient. A more efficient possibility to improve the accuracy of the estimates is to use more points on the load-displacement curve as inputs.



(a) Geometry and boundary conditions of the mesoscale model

(b) Damage distribution at the final load step

Figure 7.6 Damage variable ω at the final load step.

7.4 Mesoscale model of concrete

The final application of the parameter identification procedure is performed for a mesoscale model of concrete. In this case, the determination of the material parameters is complicated by the fact that, in general, only macroscopic experiments are available, which are possibly influenced by several mesoscale parameters in a similar way, e.g. the tensile strength of the matrix and the interfacial transition zone might influence in a similar way the macroscopic tensile strength. The aim of this example is not to determine all material parameters of the mesoscale model, but to demonstrate the applicability of the proposed parameter identification procedure. For that reason, the compressive strength of the matrix is included as a parameter to be determined, although it seems to be obvious that for the tensile test the determination of that parameter is rather erroneous.

7.4.1 Numerical model

The test used to identify the material parameters is a uniaxial tensile test. The specimen dimension was 25x25mm for the small model and 50x50mm for the large model as illustrated in Fig. 7.6(a). Lateral displacements at the top and bottom layer are not restricted and a direct displacement control is applied to obtain the load-displacement curve.

The mesoscale model consists of three components - aggregate, mortar matrix and the interfacial transition zone. Details of the geometry creation, the material models and the solution procedure can be found in chapter 4. Aggregates between 0.5 and 8mm with a total mass fraction of 0.52 are simulated. Smaller aggregates than 0.5mm where assumed to be homogenized in the matrix material, which finally gives a mass fraction of 0.7 for the aggregates.

The material parameters for the mesoscale model are the Young's modulus E^a and the Poisson's ratio ν^a of the aggregate, the Young's modulus E^m, Poisson's ratio ν^m, uniaxial tensile

Table 7.1 Range of material parameters from engineering experience.

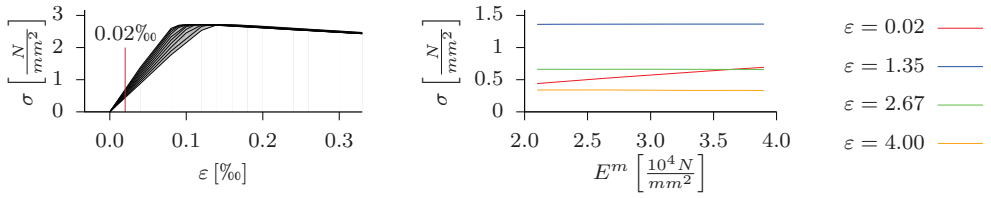
parameter	E^m $\left[\frac{N}{mm^2}\right]$	G^m $\left[\frac{Nmm}{mm^2}\right]$	f_{ct}^m $\left[\frac{N}{mm^2}\right]$	f_{ck}^m $\left[\frac{N}{mm^2}\right]$	G_{ct}^i $\left[\frac{N}{mm^2}\right]$	f_{ct}^i $\left[\frac{N}{mm^2}\right]$
min	20000	0.05	2	20	0.0125	1.125
max	40000	0.15	4	40	0.0375	3.375

strength f_{ct}^m, uniaxial and biaxial compressive strength f_{ck}^m,f_{ck2}^m and fracture energy G^m of the mortar matrix and the penalty stiffness K_p^i, tensile strength f_{ct}^i, mixture parameter α^i and fracture energy G^i of the cohesive zone model representing the interface. The Young's modulus E^a and the Poisson's ratio ν^a of the aggregates are assumed to be $E^a = 54 \cdot 10^3 \frac{N}{mm^2}, \nu^a = 0.2$ which might correspond to a granite. These parameters can be directly measured for the pure aggregate. The biaxial compressive strength f_{ck2}^m was assumed to be related to the uniaxial compressive strength by a factor of 1.16 [261]. Furthermore, a Poisson's ratio $\nu_m = 0.2$ was prescribed. The penalty stiffness K_p^i of the interface is only a numerical parameter to avoid mutual penetration of the crack faces in compression and has been set to $K_p^i = 25 \cdot 10^3 \frac{N}{mm^2}$. Shear and normal crack opening are assumed to have the same influence in the interface implying $\alpha^i = 1$. The remaining parameters E^m, f_{ct}^i, f_{ck}^m, G^m, f_{ct}^i and G^i should be determined by the parameter identification procedure described in the previous sections using the tensile test. It seems to be obvious that a tensile test is not well suited to determine the material parameters describing the compressive failure, but this will be one of the results. For other tests/other material laws it might not be straightforward to decide, if a certain test procedure is adequate to determine certain material parameters.

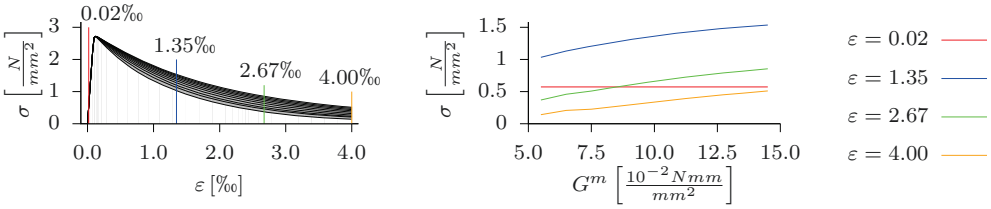
Figure 7.6(b) shows the damage variable ω at the final load step for a certain parameter set. Crack bridging can be clearly identified in the figure, where two parallel cracks have develop and finally join into a single crack.

7.4.2 Sensitivity analysis

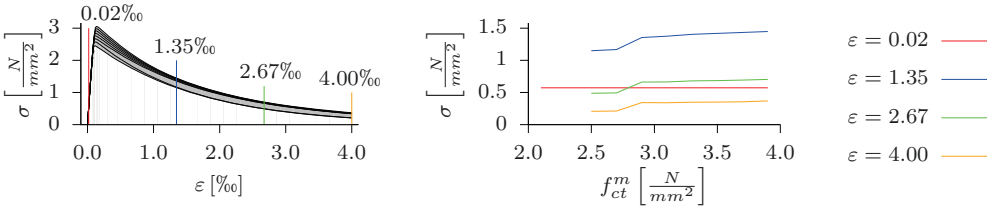
In a first approach, the sensitivity of the load-displacement curve of the tensile test with respect to the material parameters is investigated. In order to describe the influence of the material parameters on the load-displacement curve, the load at a distinct number of prescribed displacement values is extracted. All material parameters, apart from the one being investigated, are set to expected values (mean of the interval given in Table 7.1), whereas the remaining parameter is varied within the interval. These intervals have to be chosen according to engineering experience. Figure 7.7 shows the stress-strain curve (load-displacement curve normalized by thickness and length) and the stress at given strains $\varepsilon_1 = 0.02‰$, $\varepsilon_2 = 1.35‰$, $\varepsilon_3 = 2.67‰$ and $\varepsilon_4 = 4.00‰$ for the variation of a single material parameter, where ε_1 has been chosen to be located inside the elastic region, ε_4 corresponds to the maximum displacement applied and ε_2, ε_3 are distributed equally spaced in between. Note the different scaling of the x-axis for Fig. 7.7(a). Obviously, the influence of the Young's modulus E^m of the matrix is restricted to the initial elastic regime described by ε_1, whereas the stress at the other strain levels is almost independent of E^m. In fact, the only material parameter that influences the



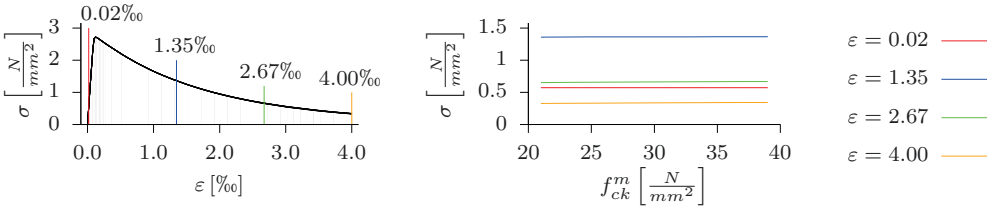
(a) Influence of E^m in the range $[2, 4] \cdot 10^4 \frac{N}{mm^2}$



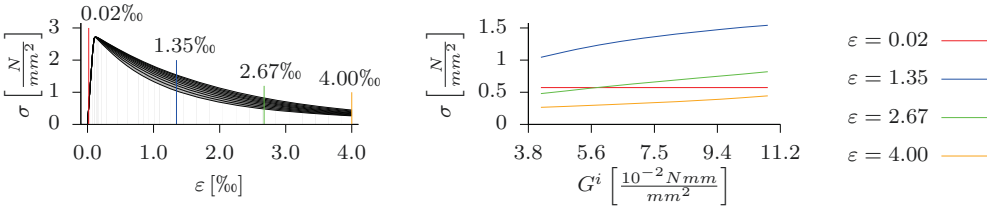
(b) Influence of G^m in the range $[0.05, 0.15] \frac{Nmm}{mm^2}$



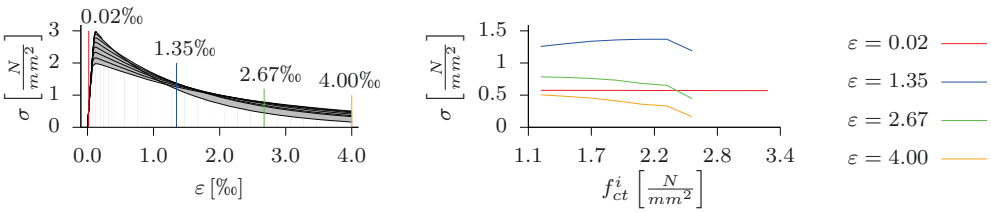
(c) Influence of f_{ct}^m in the range $[2, 4] \frac{N}{mm^2}$



(d) Influence of f_{ck}^m in the range $[20, 40] \frac{N}{mm^2}$



(e) Influence of G^i in the range $[0.0375, 0.1125] \frac{Nmm}{mm^2}$



(f) Influence of f_{ct}^i in the range $[1.125, 3.375] \frac{N}{mm^2}$

Figure 7.7 Sensitivity of the stress-strain curve with respect to the variation of a single material parameter.

stress at strain level ε_1 is the Young's modulus E^m. The influence of the fracture energy of the matrix G^m and the fracture energy of the interface G^i seems to be similar, although the variation of the fracture energy G^i primarily changes the first part of the stress-strain curve (at strain level ε_4 the curves approach each other again), whereas the fracture energy G^m influences the full range of the stress-strain curve. In a parameter identification procedure performed manually by adjusting the parameters it is not straightforward to decide, which parameter has to be adjusted. The same holds for the tensile strength of the matrix f_{ct}^m, which has a similar influence.

An interesting feature show the stress-strain curves in Fig. 7.7(c), where at ε_4 two groups with an almost identical stress level in each group can be identified. This is due to the fact that in the sensitivity analysis a constant aggregate distribution (the aggregates are for all samples at the same position) is used. However, the resulting crack path depends on the material parameters, and, by varying f_{ct}^m only slightly, the crack path might change and a strong variation of the stress-strain curve and, consequently, a discontinuous influence on the stress at $\varepsilon_2, \varepsilon_3, \varepsilon_4$ is observed. This phenomenon can be avoided by either using for each parameter set a different aggregate distribution, which actually corresponds to adding noise due to the aggregate distribution to all samples, or even better by increasing the size of the model to reduce the influence of a single aggregate position.

The compressive strength of the matrix f_{ck}^m has, as expected for a tensile test, almost no influence on the stress-strain curve. Only close to the final crack state, where due to crack bridging the principal components of the stress tensor are not aligned with the axis of the applied tensile load, a negligible influence is recognizeable.

The simulations with a tensile strength of the interface f_{ct}^i larger than the tensile strength of the matrix f_{ct}^m did not converge. This is due to the fact that in this case the interface elements representing the interfacial transition zone (ITZ) do not open and the crack has to propagate around the aggregates in the matrix. The ITZ is a weak layer around the aggregates with a reduced strength compared to the matrix. Consequently, these parameter configurations can be excluded a priori.

7.4.3 Generation of training data

From the sensitivity analysis in the previous section it is obvious that a parameter identification procedure by hand is difficult and no information about the reliability of the determined parameter set is obtained. Consequently, an automatic procedure is required. A parameter identification procedure based on Bayesian neural networks is a promising tool to circumvent the above mentioned problems. From experience, a certain range of the parameters can be specified. For the mesoscale example, this parameter range is given in Table 7.1.

Assuming a uniform distribution over the range with no correlations between the material parameters, a latin hypercube sampling can be used to obtain a training set with a certain number of samples. In the example discussed here, a latin hypercube sampling with 500 samples has been used, where the first samples where used for training and the remaining samples for testing. It is not required to use a uniform distribution, but this assures the same density of data points in the whole parameter space. Four different configurations

where investigated. First of all, a large and a small model with an edge length of 50mm and 25mm was used. In the large model, the influence of a single aggregate is not as prominent as in the small model. For each of these models, two strategies for the generation of the mesoscale models was used. In the first approach, the geometry of the mesoscale model is identical for all samples, and only the material parameters are varied according to the latin hypercube sampling. In the second approach, a different mesostructure was used for each set of material parameters. This has the advantage of including the stochastic character of the mesostructure in the simulation, which actually corresponds to adding a certain noise level due to the stochastic aggregate distribution to the parameter identification procedure.

For all these training samples, each with a different set of parameters, the numerical simulation is performed. The stress-strain curves can be extracted and, similar, the stress at the strain levels $\varepsilon_{i=1..N}$. A Bayesian neural network is created with the stress level at strain levels $\varepsilon_{i=1..N}$ as input (N inputs) and the material parameters as outputs - in this example E^m, f_{ct}^i, f_{ck}^m G^m, f_{ct}^i and G^i. The number and location of ε_i is userdefined and should be chosen in order to represent the most prominent features of the experimental curve. If the stress level in the stress-strain curve is not uniquely defined by the strain level (e.g. problems with a snap back), the strain value can be replaced by a parameter describing the integral along the curve from the origin.

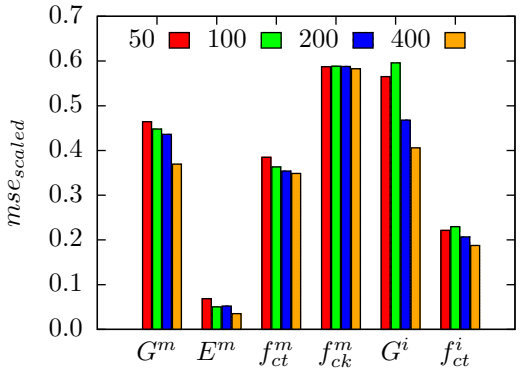
7.4.4 Network architecture and training

The network architecture should be defined by the user. In these examples, networks with a single hidden layer and sigmoidal activation functions and an output layer with linear activation functions are used. Theoretically, the number of neurons can be chosen to be large, since as shown in the previous examples, the Bayesian approach offers an appropriate regularization to avoid overfitting. However, the computational time for large networks increases (the function/gradient evaluations take longer and, additionally, the number of steps in the conjugate gradient approach increases). Consequently, it is beneficial to limit the number of neurons. A practical approach is to determine the number of neurons such as to obtain as much free parameters in the network (weights, biases, hyperparameters) as training information (number of training samples multiplied by number of outputs). In the current example, a hidden layer with 30 neurons has been used.

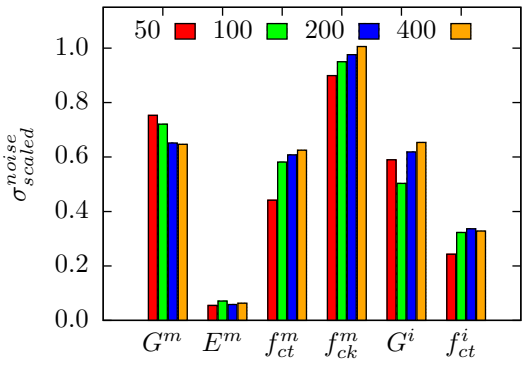
The training of the Bayesian neural network is performed using the procedure described in section 6.3 with an iterative procedure for the update of the weights and biases on the on hand, and the hyperparameters on the other hand.

7.4.5 Influence of the number of training samples

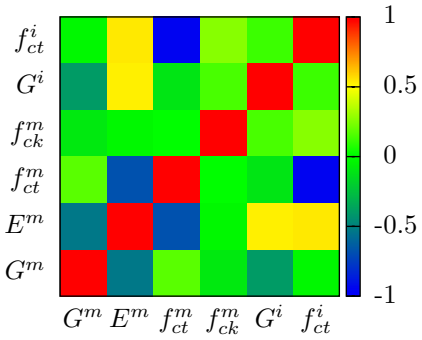
First of all, the influence of the number of training samples is evaluated. In Fig. 7.8 the scaled mean square error (mse divided by the mean of the material parameters to obtain a comparable magnitude) and the noise variances σ_i are shown for a varying number of training samples using the large model with a varying aggregate distribution and 10 points on the load-displacement curve equally spaced in the interval $[0.0005, 0.1]$, which corresponds to maximum



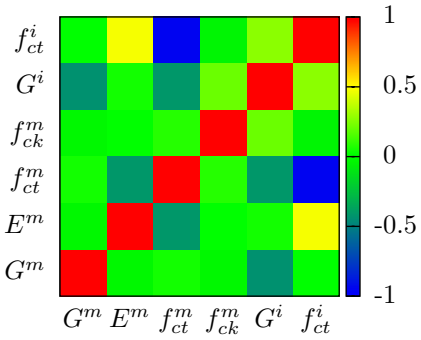
(a) Scaled mean square error of test set



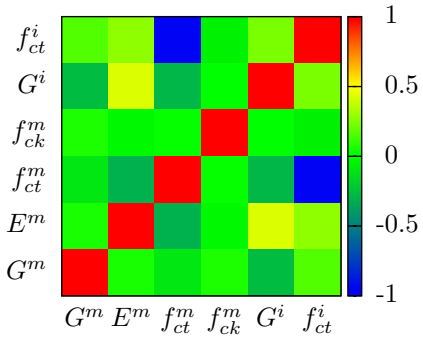
(b) Noise of training set



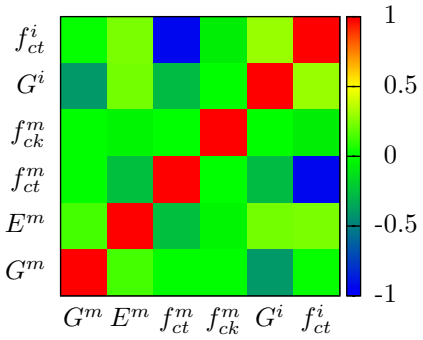
(c) Correlation noise 50 samples



(d) Correlation noise 100 samples



(e) Correlation noise 200 samples



(f) Correlation noise 400 samples

Figure 7.8 Influence of the number of samples on the results using the large model with a varying aggregate distribution for each set of parameters and 10 input points on the load-displacement curve.

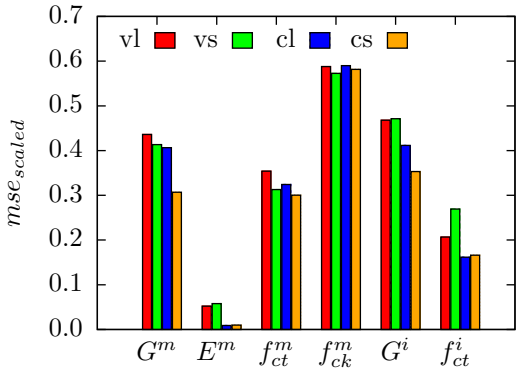
strains between 2‰ for the large and 4‰ for the small model. It is realized that by increasing the number of samples, only a minor increase of the accuracy it obtained, which indicates that the calculations with 200 samples used in the following investigations are sufficient for this problem. It is further observed that the Young's modulus of the matrix can be accurately determined, which coincides with expectations. The identification of the compressive strength is almost impossible, indicated by a scaled standard deviation of the noise being almost one. Additionally, the interfacial tensile strength can also be reasonably well determined. From the correlation coefficients it is further evident that the tensile strength of the matrix and the tensile strength of the interface are negatively correlated. From a practical perspective, this can be explained by their similar influence on the load-displacement curve - if one tensile strength is increased, the other one has to be decreased in order to obtain the same response. Fracture energy of matrix and interfacial transition zone are also negatively correlated, since, for a single load-displacement curve, the overall energy dissipated in the numerical model has to be constant.

7.4.6 Influence of the model geometry

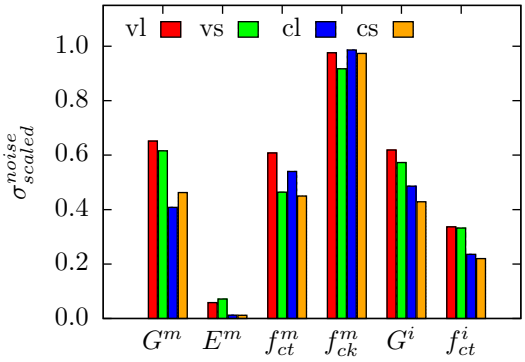
In the second investigation, the influence of the mesomodel itself is analyzed. Figure 7.9 illustrates the results for the four investigated models with a constant/varying aggregate distribution per training sample and a small/large model.

The Young's modulus can be determined for the models with a constant aggregate distribution much more accurate compared to the models with a varying aggregate distribution for each training sample. This is explained by the fact that the stochastic aggregate distribution leads to different homogenized initial stiffnesses and, consequently, additional noise is added to the training data. This additional noise is automatically identified in the parameter identification procedure. Similar conclusions can be drawn for all other parameters. However from a practical point of view, the simulation with a varying aggregate distribution is more realistic, since in a numerical simulation, the aggregate distribution can not be fixed. If e.g. the geometry is only slightly varied to simulate another model, a different aggregate distribution is obtained. As a result, the parameter estimation procedure with the varying aggregate configuration is preferable.

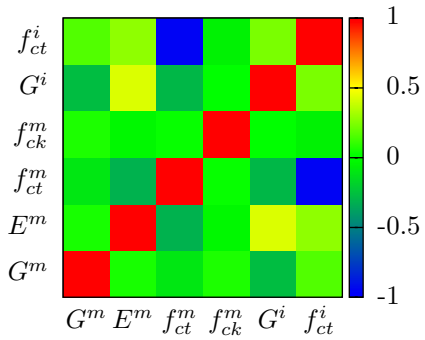
A comparison between the small and the large models shows that, in general, the small model is more accurate than the large model. This is due to the fact that, as a result of the constant maximum displacement applied on the top of the specimen, the maximum strain is larger and thus the softening branch is much more developed in the small model. Comparing e.g. in the sensitivity analysis in Fig. 7.7 the influence of the fracture energy of the matrix and the interface, it is obvious that in the strain range $0 - 0.2$‰ as used in the large model both parameters have the same influence (which is also indicated by the strong negative correlation illustrated in Fig. 7.9), whereas only in the strain range $0.2 - 0.4$‰ the different influence of both parameters becomes prominent and, consequently, the correlation in the parameter estimate is reduced.



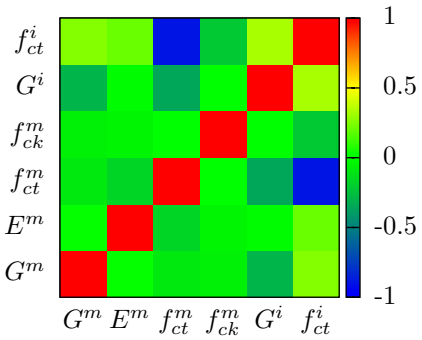
(a) Scaled mean square error of test set



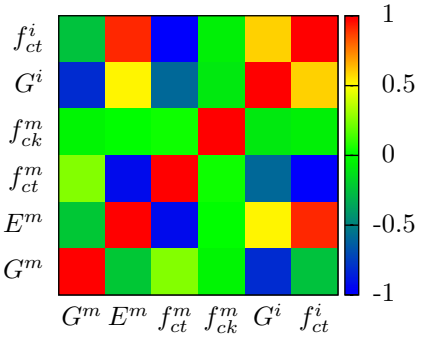
(b) Noise of training set



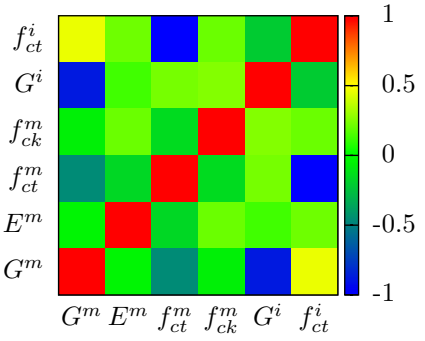
(c) Correlation noise large var



(d) Correlation noise small var

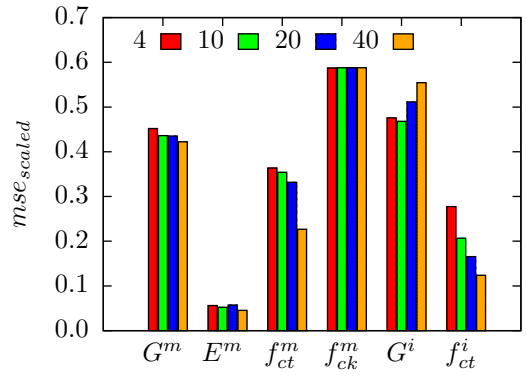


(e) Correlation noise large const

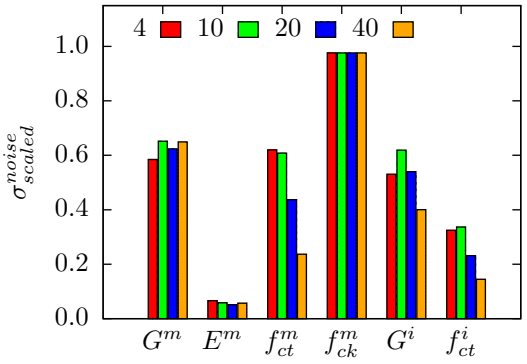


(f) Correlation noise small const

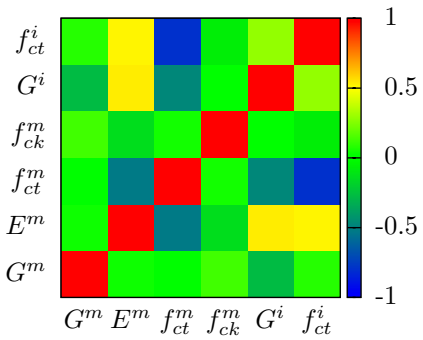
Figure 7.9 Influence of the model geometry (large(l)/small(s), varying(v)/constant(c) aggregate distribution per sample) using 200 samples and 10 input points on the load-displacement curve.



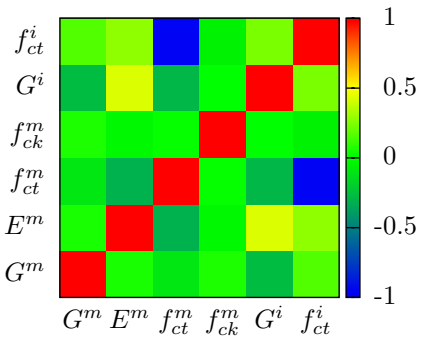
(a) Scaled mean square error of test set



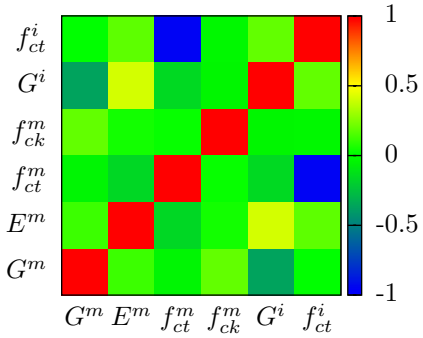
(b) Noise of training set



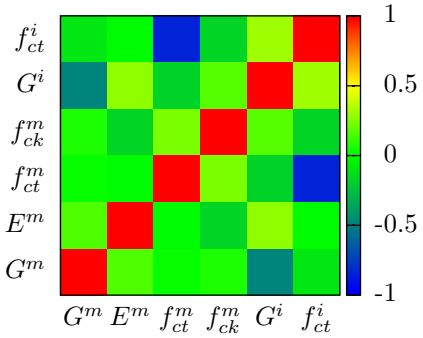
(c) Correlation 4 inputs



(d) Correlation 10 inputs



(e) Correlation 20 inputs



(f) Correlation 40 inputs

Figure 7.10 Influence of the number of input points on the load-displacement curve using 200 samples and the large model with a varying aggregate distribution for each set of parameters.

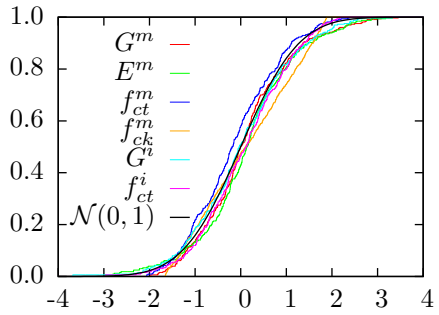


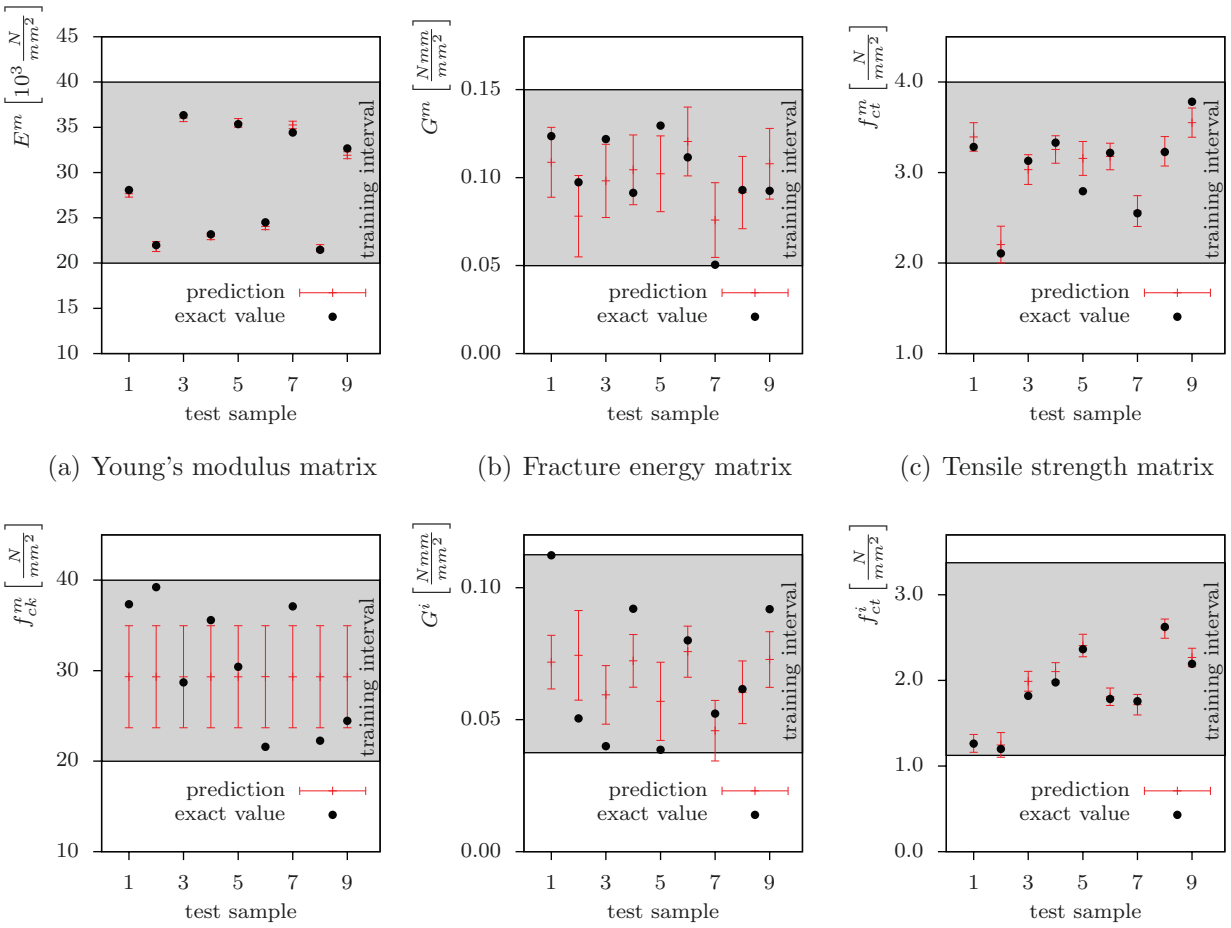
Figure 7.11 Cumulated distribution function of the error in the parameter estimate divided by the confidence interval using the large model with a variable aggregate distribution, 200 training samples and 10 points on the load-displacement curve for the test samples.

7.4.7 Influence of the number of points in the load-displacement curve

In a last setup, the number of points on the load-displacement curve and, consequently, the number of input points of the neural network is investigated. Comparing the results illustrated in Fig. 7.10 with the previous example, where the number training samples has been increased, it can be stated that the accuracy of the parameter estimates can be drastically increased by considering more points on the load-displacement curve, whereas the increase of the number of training samples has only a minor role. This is also in accordance with the example of the sin-function in section 6.4, where similar results could be obtained. It is further to be noted that only the accuracy of the parameter estimates for the tensile strength of the matrix and the interface as well as the fracture energy of the interface could be increased by an increased number of input points, whereas almost no influence on the Young's modulus and the fracture energy of the matrix is observed. This is certainly due to the fact that, by using more points, the region around the peak of the load-displacement curve is much better represented in the training samples of the network and, consequently, the approximation accuracy increases. Additionally, this hypothesis is supported by a closer examination of the correlation coefficients between tensile strength of the matrix and Young's modulus. For the setup with only four points on the load-displacement curve, a high negative correlation is obtained, whereas in the other limit using 40 input points the correlation is negligible. This is due to the fact that in a setting with 4 points, the first point is in the elastic region and the second point is already far behind the peak point and, consequently, the peak point is not accurately represented in the model. Using more input points, the region around the peak point is better resolved and the accuracy of the parameter estimates that influence this region can be increased.

7.4.8 Verification of confidence interval

In order to verify the methodology, the network response is calculated for the test set (the remaining samples of the 500 latin hypercube samples, which have not been used for training the Bayesian neural network). The cumulated density function cdf of the error between the approximated parameter and the real value divided by the confidence interval for the



(a) Young's modulus matrix

(b) Fracture energy matrix

(c) Tensile strength matrix

(d) Compressive strength matrix

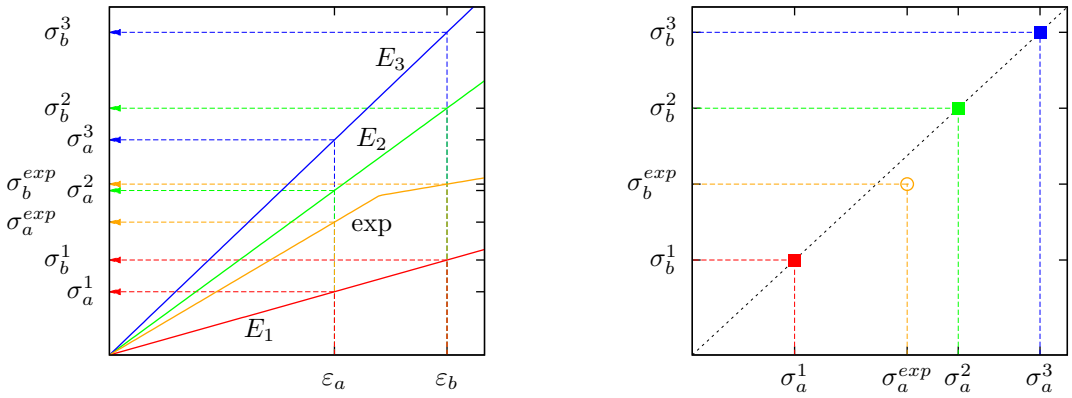
(e) Fracture energy interface

(f) Tensile strength interface

Figure 7.12 Comparison of predicted parameter distribution with the correct values for the large mesh with variable aggregate distribution using the first 9 test samples.

approximation is given exemplary for the large model with a variable aggregate distribution in Fig. 7.11. It is verified that this function is close to the normalized Gaussian distribution and demonstrates that the approximation of the response with a Gaussian distribution is justified.

The exact material parameters of 9 test samples are compared to the approximated parameters with their standard deviation in Fig. 7.12 for the setup with 200 samples, a variable aggregate distribution and 40 input points on the load-displacement curve. Obviously, the estimate for the Young's modulus is very accurate. For the tensile strength of the matrix and the interface, the error is in the range of 20% which is, compared to the errors in experimental verifications, still acceptable. It is further to be noted that in the case of an overestimation of the tensile strength of the interface, the corresponding tensile strength of the matrix is underestimated, which proves their correlation and the difficulty to determine these parameters separately. The estimate for the compressive strength of the matrix is just the mean of the training data, and the estimate is constant regardless of the input parameters. This is an advantage compared



(a) Load-displacement curves (b) Support points of metamodel

Figure 7.13 Problem of extrapolation with the Bayesian neural network, if the numerical model is not able to fit the experimental data.

to standard neural networks used for parameter identification, since there, the approximation is likely to oscillate and a regularization of a single output is difficult to realize. In order to improve the accuracy of the parameter identification for this example, additional tests could be added, e.g. a compression test, in order to determine the compressive strength and increase the accuracy of the other parameter estimates.

7.5 Limitations of the method

If the Bayesian neural network is used for the determination of material parameters from an experimental setup, two conditions have to be verified. First of all, the identified parameters should be in the range of the training intervals. Otherwise, the procedure should be repeated with an adapted interval for the training data. A second point, even more important, is the fact that the numerical model must be capable of reproducing the experimental data set, i.e. there must be at least one set of parameters that can represent the experimental data set. Otherwise, an extrapolation with the neural network approximation is obtained. This is illustrated in Fig. 7.13. Assume a linear elastic model with a single parameter E to be determined. Performing the described procedure, the parameter E is sampled within a prescribed range and the corresponding load-displacement curves are obtained. In Fig. 7.13, the curves are exemplarily plotted for the parameters E_1, E_2 and E_3. Defining two strain states, at which the load-displacement curve is to be evaluated, gives the training data as illustrated in Fig. 7.13(b). All the points of the training data are located on a straight line, which implies that σ_b^i/σ_a^i is a constant. Assume furthermore that an experimental curve with a bilinear distribution is given as illustrated in Fig. 7.13(a). The corresponding point in Fig. 7.13(b) is not located on the straight line. Since training data is only available on this straight line, an extrapolation has to be performed, i.e. the point of interest is outside the interval included in the training data. Extrapolation with neural networks has always to be handled with precaution and should be avoided as often as possible. In the Rayesian framework, the experience of the author is that the confidence interval of the parameter

estimate dramatically increases, since the eigenvalues of the Hessian matrix approach zero and the second term in Eq. (6.60) becomes large. In this context, it is therefore recommended to start with a complex numerical model, and then remove the model parameters that have no influence.

Chapter 8

Scale coupling and simulation of material behavior via metamodels

8.1 Introduction

Simulations on small scales allow for the consideration of physical phenomena, which can only be incorporated in a macroscale simulation in a smeared approach. However, this approach also increases the computational effort dramatically. In order to apply these models to real structures, two approaches are investigated in this work. In the first approach, a complex macroscopic material model can be used, whose parameters are determined by mesoscale simulations using the parameter identification procedure described in the previous chapter. Another possibility are multiscale simulations, where two scales are used simultaneously in a single model as e.g. in [262, 263], where, if a certain criterion is fulfilled, an adaptive refinement to lower scales is performed. Other authors perform simulations in parallel with models on both scales [264, 265, 266], which a transfer of information between these models.

The parallel computation on both scales is very time consuming, since, for each evaluation of stresses and strains for a macroscopic point, a full mesoscale simulation has to be performed. An alternative, presented in this chapter, is the approximation of the constitutive behavior on the mesoscale by a neural network. From a certain number of mesoscale experiments, the mesoscale model is approximated by a neural network. By using homogenization techniques, the constitutive relation for a macroscopic point is obtained. In general, the complexity of the required neural network is determined during the training such that the minimal network is used that can accurately represent the training data. Special attention is turned to the inclusion of a softening response of the material. Close to the peak load, localization and damage accumulation occurs in certain parts of the structure, whereas other parts start to unload. This unloading has to be modeled within the framework of the neural network. Furthermore, the size of the localization zone is, from an experimental point of view, independent of the mesh size, which should accurately be represented by the constitutive model. A further advantage of the method is that the training data can directly be extracted from experimental data without the need for numerical mesoscale simulations.

The first application of neural networks to material modeling was presented in [267, 268], where the material relation between strains and stresses of concrete under monotonic biaxial loading under plane stress assumptions was approximated by a neural network. In [269], the return mapping procedure in a plasticity formulation was replaced by a neural network which reduced the computational time substantially. Furukawa used neural networks to model a viscoplastic material [270], where the network was used to approximate the evolution

of viscoplastic strain and history variables. Lefik demonstrated the application of neural networks to elasto-plastic material models and discussed possibilities to reduce the influence of the size of the strain increment [271]. In [272, 273], constitutive models of geomaterials using neural networks are obtained from experimental tests with nonuniform stress states. An efficient tool in this concept is autoprogressive training, where the full constitutive model is learned in an iterative procedure by comparison between output of the neural network and training data [274]. In [275], the nonlinear response of periodic unit cells is used to train a network that can model the homogenized response on the macroscale. Furthermore, Hashash used a neural network to approximate viscoelastic material behavior [276].

In general, the curse of constitutive modeling with neural networks is the fact that the complexity of the function space is difficult to control and certain restrictions are difficult to fulfill. A straightforward example is the modeling of the yield surface in a plasticity formulation using neural networks, which has to be a convex function in order to obtain a unique solution. This convexity property is difficult to enforce with a neural network. Another crucial point is mesh and step length independence of the results. For a different load step size or a finer mesh, the results should be almost equal. The latter is especially important, if softening is included in the material response, since the localization zone is often strongly related to the mesh size.

Many of the authors applying neural networks for the approximation of the response of a material model use the strain tensor $\boldsymbol{\varepsilon}^k$ as input and approximate the corresponding stresses $\boldsymbol{\sigma}^k$ [267, 268]. For history dependent material behavior, additional variables as e.g. the stress $\boldsymbol{\sigma}^{k-1}$ and the strain tensor of previous equilibrium states $\boldsymbol{\varepsilon}^{k-1} = \boldsymbol{\varepsilon}^k - \Delta\boldsymbol{\varepsilon}^k$ are considered as additional input parameters. The number of previous equilibrium states considered corresponds to the complexity of the history dependent material response. The drawback of these models is the dependence of the neural network approximation on the strain increment. When the material model is used in a complex implicit Newton-Raphson scheme to solve for the macroscale response, the strain increment is controlled by the applied load increment which has to be adapted according to the training interval. In order to assure a robust solution, the network must be trained for a large range of $\Delta\boldsymbol{\varepsilon}^k$. Another important aspect is the number of training samples required to obtain a network with accurate generalization behavior. Using only one history module which corresponds to the stresses and strains of the previous equilibrium state requires $3d$ input parameters, where d is the dimension of the stress vector $(\boldsymbol{\varepsilon}^{n-1}, \boldsymbol{\sigma}^{n-1}, \boldsymbol{\varepsilon}^n)$. Since the number of required training samples grows exponentially with the dimension of the input vector, already a plane strain model with 5 training samples per dimension requires $5^{3 \cdot 3} = 1953125$ training samples.

In this work, a different approach is proposed. The history dependent material response is considered by additional history variables, which are related to the maximum strain ever reached for a certain point in the loading process. This definition is based on the assumption that only the maximum strain itself - and not the path - is important for the material response.

The proposed algorithm can be decomposed into two stages. The first stage includes the calculation of the current history parameter as a function of the new total strain using a metamodel, which describes the loading and unloading behavior. This metamodel can be interpreted as the equivalent of a yield function in an elasto-plastic calculation. It is modeled

by a support vector machine, since the training samples only include the binary information loading/unloading represented by ± 1. In a second stage, the stresses are calculated as a function of the current strain and the current history variables using a multilayer perceptron. The individual steps are discussed in the next section separately.

8.2 Generation of training data from mesoscale simulations

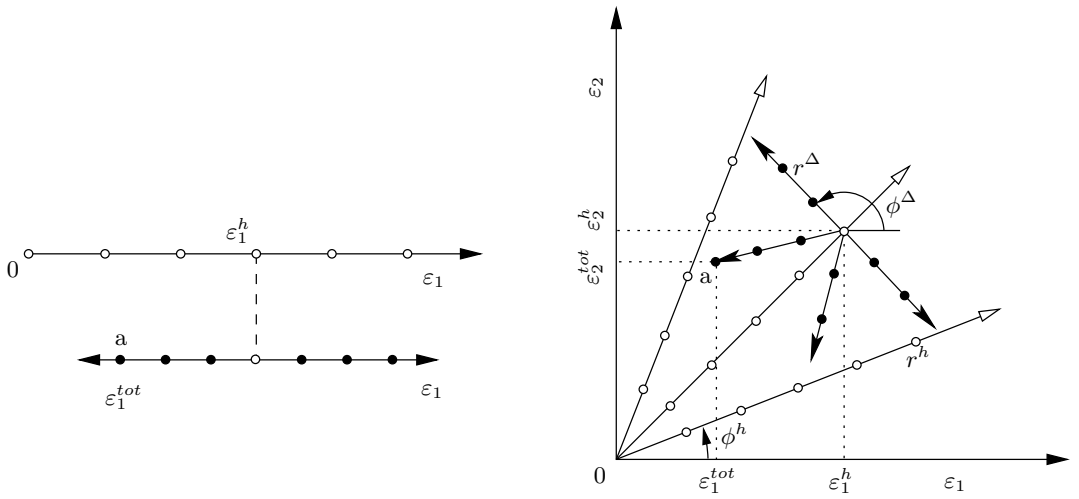
The training data for the approximation of the material response is extracted by homogenization from results of numerical simulations of models with a high degree of detail and, consequently, a high computational effort - e.g. a mesomodel whose response is included as a homogenized material formulation in a macroscopic simulation. In the following, the detailed model is named the mesoscale model, whereas the model, which incorporates the detailed model as a material formulation on the integration point level, is named the macroscale model.

In order to generate the training data, a certain number of mesoscale simulations under different loading conditions is performed. Since softening is included in the mesoscale model, a displacement controlled approach is used. The input to the network is the homogenized strain, which is related to the applied displacements, whereas the output is the homogenized stress, which is related to the reaction forces on the boundary of the model.

For standard applications using neural networks, it is in general recommended to select the training data randomly, e.g. by Monte Carlo or Latin Hypercube sampling. Due to numerical issues (e.g. convergence of the Newton-Raphson iteration), the final load/displacement cannot be applied to the mesoscale model in a single step, but must be decomposed into several increments. For all these increments an equilibrium solution of the system is found, which can be used as training data for the network. In order to reduce the number of mesoscale simulations to a minimum, a DOE scheme (design of experiments) is used. This is illustrated for problems with one and two strain components in Fig. 8.1. Note that in a standard solid mechanics formulation only 1,3,4 and 6 strain components are possible. The strains can in a similar way be interpreted as the normal and tangential crack opening in a formulation for an interface element, but in order to make the notation general they are denoted as strain components ε_i. The example of interface elements is only added for illustration purposes, since, for more than two strain components, the graphical representation is difficult, and for one strain component the procedure reduces to loading up to a certain maximum strain ε^h and from there increase/decrease of the strain to obtain ε^{tot}. For the interface formulation, the inputs to the network are the total strains ε_1^{tot} and ε_2^{tot} with the corresponding history variables ε_1^h and ε_2^h. In a first step, the range of the maximum strain r^h, the number of angles ϕ^h and the number n^h of subdivisions of the radius r^h is defined. For each angle ϕ^h, the applied displacement is increased with a constant increment until $i = n^h$:

$$\begin{pmatrix} \varepsilon_1^h \\ \varepsilon_2^h \end{pmatrix} = \begin{pmatrix} \frac{i}{n^h} r^h \cos(\phi^h) \\ \frac{i}{n^h} r^h \sin(\phi^h) \end{pmatrix} \qquad i = 1..n^h. \tag{8.1}$$

At this stage, the inelastic energy $W^{inel}(\boldsymbol{\varepsilon}^h, \boldsymbol{\varepsilon}^h)$ stored in the system, which is the total energy subtracted by the elastic energy, is determined. From this state, a second loading step



(a) 1-D formulation

(b) 2-D formulation with interface elements

Figure 8.1 Generation of training samples for a formulation with a) one and b) two strain components

is performed in a similar way. The angles ϕ^Δ, the radius r^Δ and the number of subdivisions n^Δ are prescribed

$$\begin{pmatrix} \varepsilon_1^{tot} \\ \varepsilon_2^{tot} \end{pmatrix} = \begin{pmatrix} \varepsilon_1^h \\ \varepsilon_2^h \end{pmatrix} + \begin{pmatrix} \frac{j}{n^\Delta} r^\Delta \cos(\phi^\Delta) \\ \frac{j}{n^\Delta} r^\Delta \sin(\phi^\Delta) \end{pmatrix} \quad j = 1..n^\Delta, \tag{8.2}$$

and the inelastic energy $W^{inel}(\boldsymbol{\varepsilon}^{tot}, \boldsymbol{\varepsilon}^h)$ is determined, where the first parameter is the current strain state and the second parameter corresponds to the state of the load applied in the previous loading regime. If, for a strain state $(\boldsymbol{\varepsilon}^{tot}, \boldsymbol{\varepsilon}^h)$, the inelastic energy in the system increases compared to the history state $(\boldsymbol{\varepsilon}^h, \boldsymbol{\varepsilon}^h)$, this direction, represented by the variables r^Δ, ϕ^Δ, is interpreted as loading. Consequently, this direction is not followed any further, because a further step in this direction only further increases the inelastic energy and is therefore also in loading. As a result, the number of required mesoscale simulations is reduced. In higher dimensional spaces, the principal components and the rotation angles are prescribed with its maximum values and the number of subdivisions. For a plane strain analysis, this results in two principal strains and one rotation angle, whereas for a full 3D analysis three principal strains and three Euler angles are considered. These strains are transformed to kinematic homogeneous boundary conditions of the mesoscale model [277]. This is illustrated for a plane strain situation in Fig. 8.2. Additional effort is required in the case of elastic loading in the mesoscale model, corresponding to $W^{inel}(\boldsymbol{\varepsilon}^{tot}, \boldsymbol{\varepsilon}^h) = W^{inel}(\boldsymbol{\varepsilon}^h, \boldsymbol{\varepsilon}^h) = 0$. In that case, the history loading state has no influence on the response

$$\boldsymbol{\sigma}(\boldsymbol{\varepsilon}^{tot}, \boldsymbol{\varepsilon}_1^h) = \boldsymbol{\sigma}(\boldsymbol{\varepsilon}^{tot}, \boldsymbol{\varepsilon}_2^h) \;\; \forall (\boldsymbol{\varepsilon}_1^h, \boldsymbol{\varepsilon}_2^h) : \; W^{inel}(\boldsymbol{\varepsilon}^{tot}, \boldsymbol{\varepsilon}_i^h) = 0. \tag{8.3}$$

A straightforward approach would be to interpret this as unloading, but this procedure has the disadvantage that, at the transition from the unloading to the loading regime, the history strain is discontinuous (in the elastic regime it is zero and in the inelastic regime it corresponds

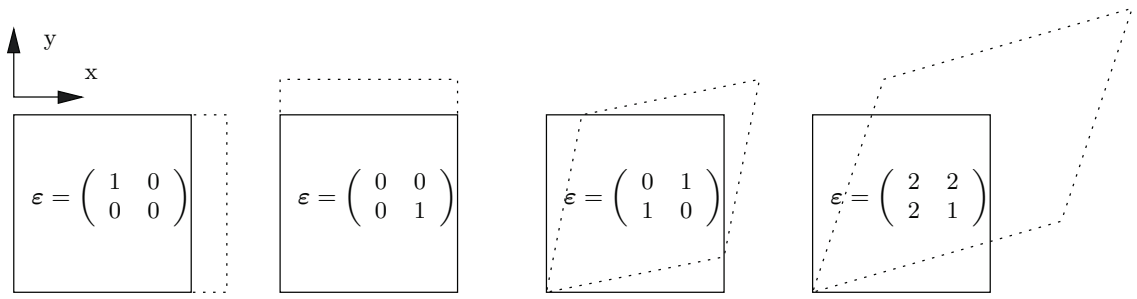


Figure 8.2 Kinematic homogeneous boundary conditions for different prescribed homogenized strain tensors under plane strain conditions

to the current total strain). As a consequence, the metamodel approximating the stress state as a function of the strain and history strain will also produce a discontinuous stress state, which finally poses problems in the global iteration using a Newton-Raphson scheme.

In the following section it is demonstrated that a modified criterion is advantageous. In the elastic regime, the norm of the strain tensor is used as a loading criterion. It is assumed that loading takes place if the norm of the strain tensor increases. The overall algorithm is illustrated in Fig. 8.3. The corresponding training data set is schematically illustrated in Fig. 8.4 with $n^h = n^{\Delta} = 6$ for a 1D-tension test with Mises plasticity. Based on these training data, a loading/unloading metamodel with the input parameters $(\boldsymbol{\varepsilon}^{tot}, \boldsymbol{\varepsilon}^h)$ and the output loading($f^t = 1$) or unloading($f^t = -1$) is created. Support vector machines have been used for the approximation, since they can efficiently handle binary data.

A second network - the stress/strain network - approximates for all points $(\boldsymbol{\varepsilon}^{tot}, \boldsymbol{\varepsilon}^h)$, which are either in the unloading regime ($f^t = -1$) or in the elastic domain, the corresponding stress tensor

$$\boldsymbol{\sigma} = f^{\sigma}(\boldsymbol{\varepsilon}^{tot}, \boldsymbol{\varepsilon}^h). \tag{8.4}$$

Multilayer perceptrons as well as radial basis functions have been investigated as network architectures, but slightly better results with respect to the accuracy and the smoothness of the approximation could be obtained with the multilayer perceptron, which was finally used in this work.

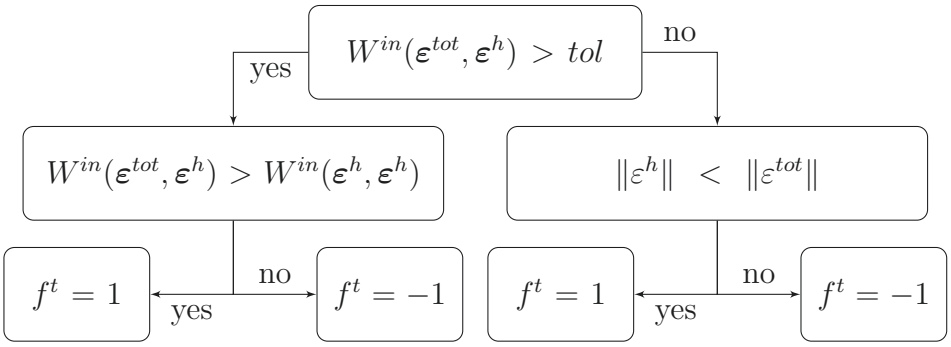


Figure 8.3 Calculation of the loading function

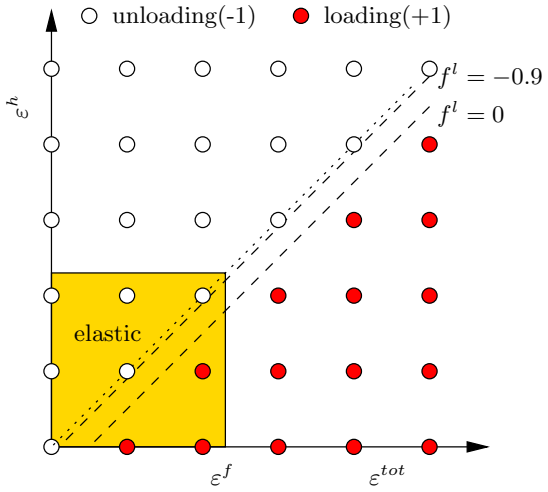


Figure 8.4 Training data for the loading/unloading metamodel

Softening is included in the mesomodel and, as a result, the homogenized response is a function of the specimen dimension. This can be illustrated for a simple 1D tension test. The zone of localization and, consequently, the dissipated energy is independent of the size of the specimen. As a result, the homogenized dissipated energy per length is linearly related to the length of the test specimen. As a result, the dimension of the specimen has to be included as additional parameter in the training process for both networks. The dependence on the specimen dimension is always considered in the following sections, but sometimes omitted in the notation to improve the clarity of the formulas.

The maximum number of mesoscale simulations N^T is given by

$$N^T = (n^h n^\Delta)^d n^l, \tag{8.5}$$

where d is the number of strain components considered in the mesomodel and n^l is the number of investigated specimen dimensions. The results of these mesoscale simulations are used to train the two metamodels. The loading/unloading metamodel is trained with input parameters $(\boldsymbol{\varepsilon}^{tot}, \boldsymbol{\varepsilon}^h)$ and the output ± 1 according to Fig. 8.3. The parameter R in Eq. (6.125), which describes the influence radius of the kernel functions, is determined a priori in order to ensure an overlapping of the support between neighboring support points. The input is scaled to the interval $[-1, +1]$ and the radius is set to $R = 1/3$.

8.3 Stress calculation for a material point on the macroscale

In a standard finite element software, a trial strain and the history variables of the last equilibrium set are given as inputs to the material formulation, which determines the new stress state and the new history variables corresponding to this stress state. In this approach, a two phase procedure is proposed. In a first step, the new history variables are determined using the loading/unloading metamodel, so that the strain state $(\boldsymbol{\varepsilon}^{tot}, \boldsymbol{\varepsilon}^h)$ is in the unloading domain (or on the border between loading and unloading), which is comparable to the return

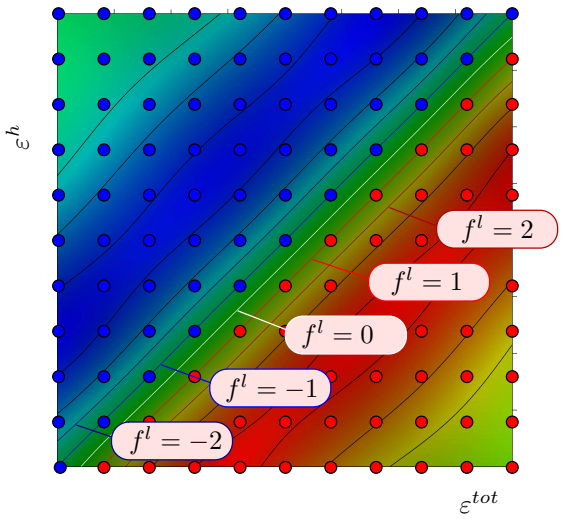


Figure 8.5 Approximation of the loading function using the support vector machine

mapping procedure in a plasticity formulation. In a second step, using this new history variables and the total strain state, the stresses are calculated from the stress/strain network.

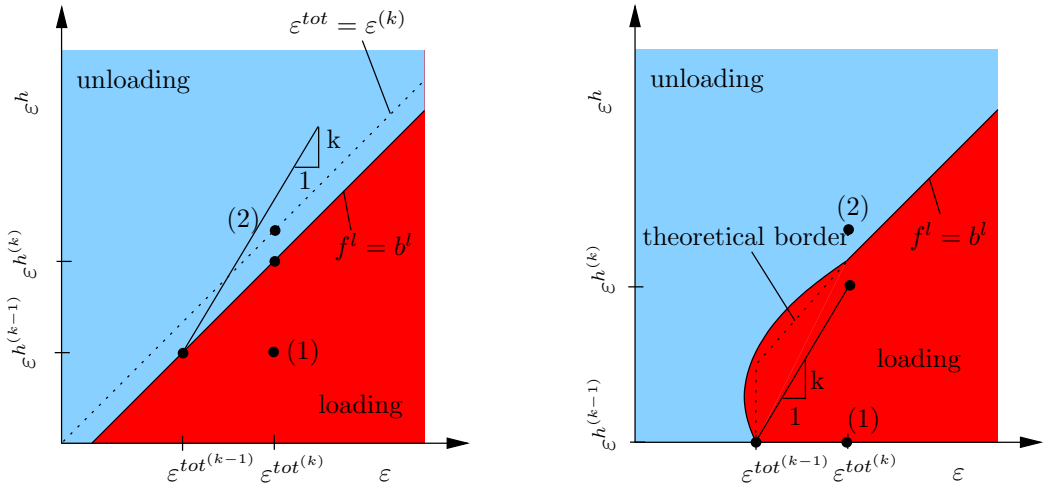
8.3.1 Loading/Unloading condition

The approximation of the loading function for a 1D problem using support vector machines is illustrated in Fig. 8.5. The isolines for $f^l = \{-2, -1, 0, 1, 2\}$ are highlighted. From a theoretical point of view, the border between loading and unloading for this simple example should be described by $\boldsymbol{\varepsilon}^{tot} = \boldsymbol{\varepsilon}^h$. Due to the definition of loading/unloading for the training samples, this line almost coincides with the isoline $f^l = -1$, but for numerical reasons there might be slight variations. Consequently, the border between loading and unloading is described by an isoline $f^l = b^l$, where b^l is a numerical parameter that is close to minus one as illustrated in Fig. 8.4. In the current implementation, a value of $b^l = -0.95$ has been used. The general procedure for the determination of the history parameter $\boldsymbol{\varepsilon}^h$ is illustrated in Fig. 8.6. Starting at the strain state $(\boldsymbol{\varepsilon}^{tot^{(k)}}, \boldsymbol{\varepsilon}^{h^{(k-1)}})$, where $\boldsymbol{\varepsilon}^{h^{(k-1)}}$ corresponds to the history variable of the previous equilibrium iteration, the loading/unloading model is evaluated (point 1). If $f^l(\boldsymbol{\varepsilon}^{tot^{(k)}}, \boldsymbol{\varepsilon}^{h^{(k-1)}}) - b^l < 0$, the strain state is considered to be in the unloading regime, which implies that the history variable $\boldsymbol{\varepsilon}^{h^{(k)}} = \boldsymbol{\varepsilon}^{h^{(k-1)}}$ is not modified within this iteration step. Otherwise, a mapping back to an unloading state by modifications of the history variable is performed

$$\Delta\boldsymbol{\varepsilon}^{tot^{(k)}} = \boldsymbol{\varepsilon}^{tot^{(k)}} - \boldsymbol{\varepsilon}^{tot^{(k-1)}} \tag{8.6}$$

$$\Delta\boldsymbol{\varepsilon}^{h^{(k)}} = \boldsymbol{\varepsilon}^{tot^{(k)}} - \boldsymbol{\varepsilon}^{h^{(k-1)}} \tag{8.7}$$

$$\boldsymbol{\varepsilon}^{h^{(k)}}_{trial} = \boldsymbol{\varepsilon}^{h^{(k-1)}} + s \cdot \min\left(1, k\frac{\|\Delta\boldsymbol{\varepsilon}^{tot^{(k)}}\|}{\|\Delta\boldsymbol{\varepsilon}^{h^{(k)}}\|}\right)\Delta\boldsymbol{\varepsilon}^{h^{(k)}}. \tag{8.8}$$



(a) Loading function using the modified criterion for loading in the elastic region.

(b) Loading function without the modified criterion for loading in the elastic region.

Figure 8.6 Modification procedure for the history variable using the loading/unloading metamodel

The variable $\|\Delta\varepsilon^{tot^{(k)}}\|$ corresponds to the norm of the strain increment of the total strain in the current iteration step (k). The increment of the history variable $\Delta\boldsymbol{\varepsilon}^{h^{(k)}}$ is defined as the difference between the current total strain and the last history state. Consequently, this definition corresponds to the assumption that in the case of loading the history state is moved in the direction of the total strain state until the loading function is equal to b^l. In the same way, the equivalent strain increment of the history variable $\|\Delta\varepsilon^{h^{(k)}}\|$ is defined. The history strain $\boldsymbol{\varepsilon}^{h^{(k)}}$ for the current iteration is calculated according to Eq. (8.8). The parameter k controls the maximum ratio between the equivalent total strain increment and the equivalent history strain increment. This restriction is used to ensure that small increments of the total strain only lead to small increments of the history strain. The problem is illustrated in Fig. 8.6(b), where in contrast to Fig. 8.6(a), all the elastic solutions are interpreted as unloading. The resulting theoretical border between loading and unloading is plotted in Fig. 8.6(b). Starting from the virgin material, the total strain is increased up to point 1, whereas the history strain remains zero. Due to the approximation with Gaussian kernel functions, the numerically obtained border between loading and unloading is curved. Without considering the factor k, a small increment of the total strain would lead to an increment of the history strain of approximately ε^{tot}. This results in convergence problems for the numerical simulation on the macroscale, since a discontinuous relation between total strain and history strain results in a discontinuous relation between total strain and total stress and, consequently, the global solution with a line search in the nonlinear Newton-Raphson iteration does not converge.

The factor s represents the distance to the border between loading and unloading. Its determination is illustrated in Fig. 8.7. The approximation of the function f^l between the function values -1 and +1 is almost linear. Furthermore, it is assumed that for $\boldsymbol{\varepsilon}^{tot^{(k)}} = \boldsymbol{\varepsilon}^{h^{(k)}}$, corre-

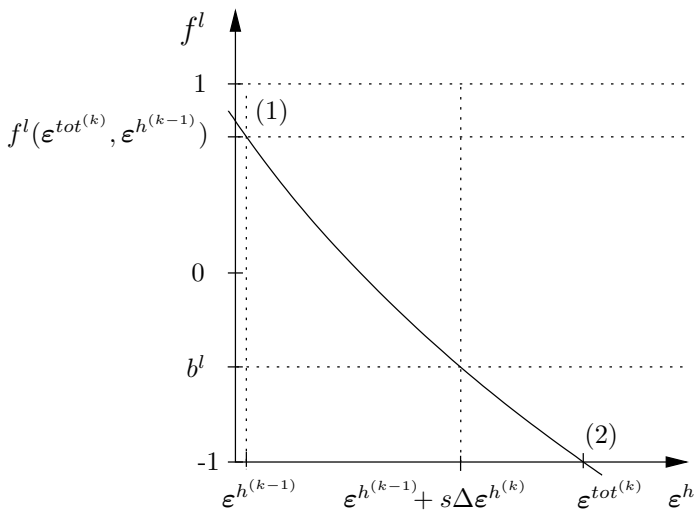


Figure 8.7 Calculation of the scaling factor s

sponding to $s = 1$, the response of the loading/unloading metamodel is almost -1. According to the theorem of intersecting lines (with $s = 0$ for $\varepsilon^{h^{(k-1)}}$), it follows:

$$\frac{f^l(\varepsilon^{tot^{(k)}}, \varepsilon^{h^{(k-1)}}) - (-1)}{1} = \frac{b^l - (-1)}{1 - s}. \tag{8.9}$$

Rearranging the terms in Eq. (8.9) gives

$$s = \frac{f^l(\varepsilon^{tot^{(k)}}, \varepsilon^{h^{(k-1)}}) - b^l}{f^l(\varepsilon^{tot(k)}, \varepsilon^{h^{(k-1)}}) + 1}. \tag{8.10}$$

In the next step, the loading function is evaluated at the trial state (2). If $f^l(\varepsilon^{tot^{(k)}}, \varepsilon^{h^{(k)}}_{trial}) - b^l < 0$, which means the trial state is in the unloading region, a Newton method is applied to exactly find the point $f^l(\varepsilon^{tot^{(k)}}, \varepsilon^{h^{(k)}}_{trial}) - b^l = 0$, where the trial state is used as starting point. Otherwise, the current history state $\varepsilon^{h^{(k)}}$ is assigned the trial state $\varepsilon^{h^{(k)}}_{trial}$. The general procedure is illustrated in Fig. 8.8.

8.3.2 Stress calculation

Once the correct history variable $\varepsilon^{h^{(k)}}$ for iteration t has been determined, the stresses can be approximated by the stress/strain network:

$$\sigma = f^\sigma(\varepsilon^{tot^{(k)}}, \varepsilon^{h^{(k)}}). \tag{8.11}$$

Note that for $f^l(\varepsilon^{tot(k)}, \varepsilon^{h^{(k-1)}}) - b^l > 0$ the sample point is in general slightly outside the region of training samples for the stress/strain network. This is due to the fact that for the training of the stress/strain network only training samples, which are either in the unloading regime or in the elastic domain are considered, which is illustrated in Fig. 8.9. Note that

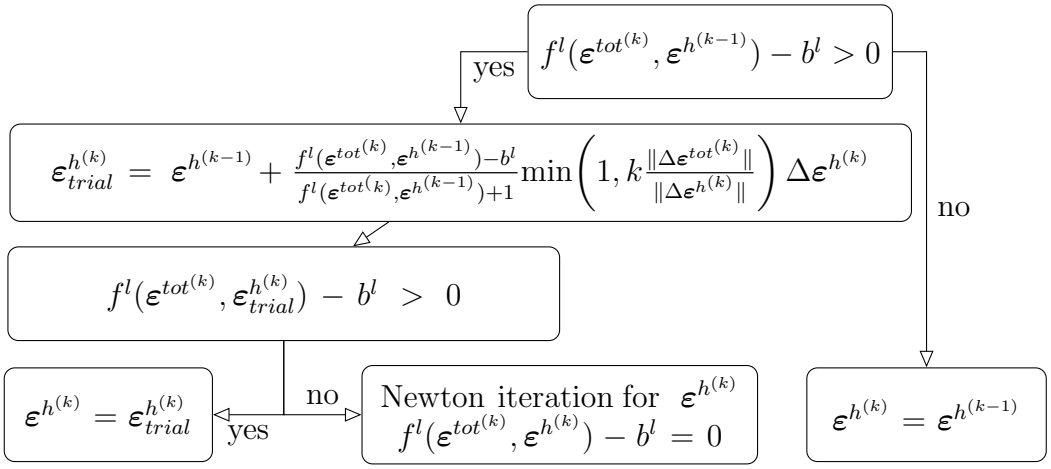


Figure 8.8 General scheme for the determination of $\boldsymbol{\varepsilon}^{h^{(k)}}$

in contrast to the loading/unloading metamodel in Fig. 8.9 all points in the elastic domain are considered. During the first investigation of the method, this restriction to training samples in the unloading or in the elastic regime was not used. Instead of deleting these samples (marked with squares in Fig. 8.9), the response was set to $f^\sigma(\boldsymbol{\varepsilon}^{tot^{(k)}}, \boldsymbol{\varepsilon}^{tot^{(k)}})$, which is always in the unloading regime. This can be compared e.g. to plasticity theory, where the current state of total strain and history variables violates the yield condition and a return mapping is performed to modify the history variables until the yield condition is satisfied. Instead of performing the modification of the history variable, the approach is to set the stress of the training sample to the stress that would be obtained after the adaptation of the history variable is performed. However, this approach results in an approximation with a discontinuous derivative. Assume for a 1D problem with perfect plasticity a strain state in the loading regime. For decreasing $\boldsymbol{\varepsilon}^{tot^{(k)}}$ the stress remains constant as long as $f^l - b^l > 0$. When entering the unloading region, a kink in the curve is obtained. Since only a limited number of training samples is used, this kink is smoothed by the neural network approximation and the overall approximation accuracy is reduced, especially in the region $\boldsymbol{\varepsilon}^{tot^{(k)}} = \boldsymbol{\varepsilon}^{h^{(k)}}$. In the elastic domain, there is no kink, since the stress response is independent of the history variable. Consequently, only the samples marked with a square in Fig. 8.9 are not considered as training samples for the stress/strain network.

In general, a full Newton-Raphson iteration is used on the macroscale and an update of the history variables $\boldsymbol{\varepsilon}^h$ is performed each time an equilibrium state is reached. The stiffness for a material point is calculated by an analytical differentiation of the metamodel.

8.4 Equivalent length

As pointed out in the introduction, the size of the mesomodel plays a crucial role in the homogenized response when softening is considered. Consequently, the equivalent length has been considered as additional input parameter of the loading/unloading and the stress/strain

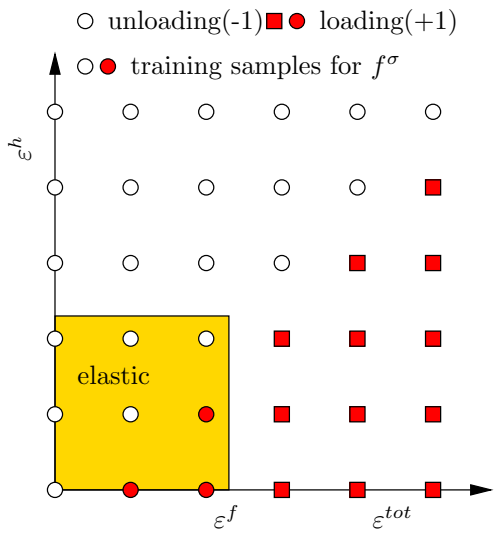


Figure 8.9 Training samples used for the stress/strain network

metamodels. For the mesoscale model, l_{eq} corresponds to the size of the test specimen, whereas for an integration point on the macroscale l^{eq} is determined as

$$l^{eq} = \begin{cases} l^{el} & \text{1D} \\ \sqrt{A^{el}} & \text{2D} \\ \sqrt[3]{V^{el}} & \text{3D,} \end{cases} \tag{8.12}$$

where l^{el}, A^{el}, V^{el} are the length, area and volume of the corresponding finite element. Extrapolation of data using neural networks is problematic. This requires that the size of any finite element in the macroscale model is contained in the interval of the smallest and the largest mesoscale model.

8.5 Numerical implementation

8.5.1 Stiffness calculation

The implementation of a the presented neural network model as a material formulation into any existing FE-code is straightforward [276]. The stress calculation can be performed as illustrated above. For the calculation of the stiffness matrices, two possibilities can be distinguished. In a first approach, the stiffness is calculated via differences. This requires $D + 1$ evaluations of the network for forward/backward differences or $2D$ evaluations for central differences, where D is the dimension of the stiffness matrix. However, this approach has the drawback to introduce the stepsize as an additional parameter. A second possibility, used in this work, is the direct differentiation of the network response with respect to the input parameters. The algorithm can be performed with a backpropagation scheme, similar to the calculation of the gradient of the objective function with respect to the weights.

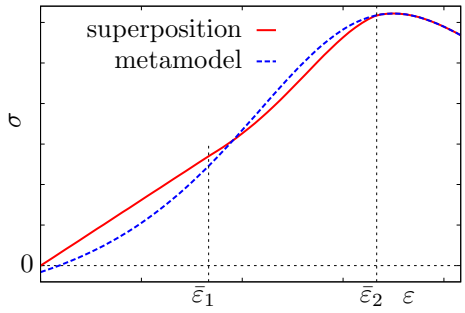


Figure 8.10 Superposition with linear elastic material law

8.5.2 Superposition with linear function

In general, it can be assumed that for a body without any applied loads, all strains and stresses vanish. Using neural networks in a material formulation, where the output $\boldsymbol{\sigma}$ corresponds to stresses and the input $\boldsymbol{\varepsilon}$ to strains (or displacement differences for an interface formulation), results in the problem that, even if there is a training point which fulfills this condition, the metamodel does not necessarily yield vanishing outputs for inputs $\boldsymbol{\varepsilon} = \mathbf{0}$. In order to represent this initial equilibrium state correctly, a superposition with a linear function is used. This corresponds to the assumption that any mesomodel can be represented close to the origin by a Taylor expansion using only the linear term. In this work, an isotropic linear elastic material law is used. The parameters of this material law can be obtained from the training samples close to the origin e.g. by a least squares fit. If, in the evaluation of the metamodel, the equivalent total strain ε_{eq} is smaller than a user-defined parameter $\bar{\varepsilon}_1$, the metamodel is replaced by the linear elastic approximation $\boldsymbol{\sigma}_{el}(\boldsymbol{\varepsilon})$. In order to obtain a smooth transition from the linear approximation to the metamodel, a transition zone with radius $\bar{\varepsilon}_2$ is introduced. The superposed response $\tilde{\boldsymbol{\sigma}}$, illustrated in Fig. 8.10 is then given by

$$\tilde{\boldsymbol{\sigma}}\left(\boldsymbol{\varepsilon}\right) = \begin{cases} \boldsymbol{\sigma}_{el}(\boldsymbol{\varepsilon}) & \varepsilon_{eq} \leq \bar{\varepsilon}_1 \\ \dfrac{\bar{\varepsilon}_2 - \varepsilon_{eq}}{\bar{\varepsilon}_2 - \bar{\varepsilon}_1}\boldsymbol{\sigma}_{el}(\boldsymbol{\varepsilon}) + \dfrac{\varepsilon_{eq} - \bar{\varepsilon}_1}{\bar{\varepsilon}_2 - \bar{\varepsilon}_1}\boldsymbol{\sigma}_{meta}(\boldsymbol{\varepsilon}) & \bar{\varepsilon}_1 < \varepsilon_{eq} \leq \bar{\varepsilon}_2 \\ \boldsymbol{\sigma}_{meta}(\boldsymbol{\varepsilon}) & \varepsilon_{eq} > \bar{\varepsilon}_2. \end{cases} \tag{8.13}$$

8.6 Example

8.6.1 Mesoscale model - generation of training data

In order to investigate the proposed algorithm, a simplified one-dimensional tension test is performed as illustrated in Fig. 8.11. The material model is a nonlocal formulation incorporating damage and plasticity. The plasticity model is based on a combination of Drucker-Prager and Rankine, whereas the damage is a function of the accumulated equivalent plastic strain. Further details of the material formulation can be found in [278] or in section 3.3. A mesh size of 50mm with a nonlocal radius of 100mm is used. The area of the element in the center

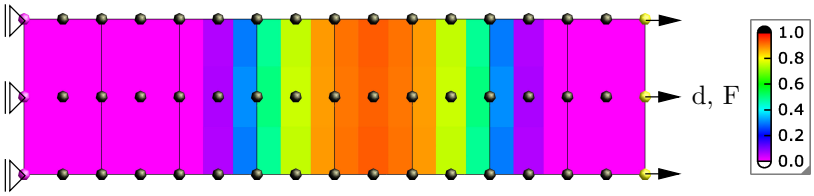


Figure 8.11 Boundary conditions and damage distribution in the mesomodel at $\varepsilon^h = \varepsilon^{tot} = 0.4‰$

is slightly reduced (3%) in order to weaken the structure and force the location of localization into this element. The tension test is repeated for specimen with a different size (from 200mm to 500mm). Furthermore, for each size, the loading and unloading process is simulated. The range of the homogenized strains and history strains, for which mesoscale simulations are performed, is in the interval [0,0.4‰], where the homogenization is simply performed by

$$\bar{\varepsilon} = \frac{\Delta d}{l}, \tag{8.14}$$

where Δd is the elongation of the model and l its length. Furthermore, this interval is divided into $n_s = 11$ or $n_s = 19$ substeps, where each substep corresponds to a training sample in the simulation. At first, the model is loaded until ε^h, then additional displacements are added as already illustrated in Fig. 8.1. For $n_s = 2$, the homogenized strain tensors which are transformed to kinematic homogeneous boundary conditions of the mesoscale model are given in Table 8.1. It is to be noted that for $\varepsilon^h = 0.0‰$ the case $\varepsilon^{tot} = 0.4‰$ is not computed, because in the ascending direction $\varepsilon^{tot} = 0.2‰$ is already determined as loading. Consequently $\varepsilon^{tot} = 0.4‰$ is also in loading, which is not added to the training samples of the stress/strain network as discussed in section 8.2. The loading/unloading network has 9 training samples, whereas the stress/strain network has only 6 training samples (only the ones in an unloading situation).

Consequently, a maximum of $n_s^2 n_l$ mesoscale simulations has to be performed, where n_l is the number of specimen sizes [200-500mm] considered. The reference solution for pure loading

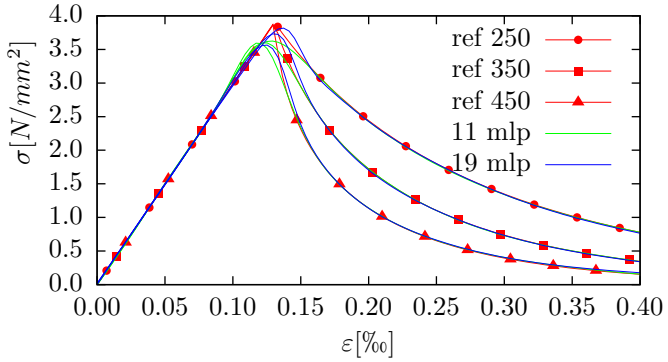


Figure 8.12 Load-displacement curve for varying size of mesoscale model(250-450mm), reference and approximation with a neural network (multilayer perceptron) for 11 and 19 training points

Table 8.1 Training data for the uniaxial tensile test

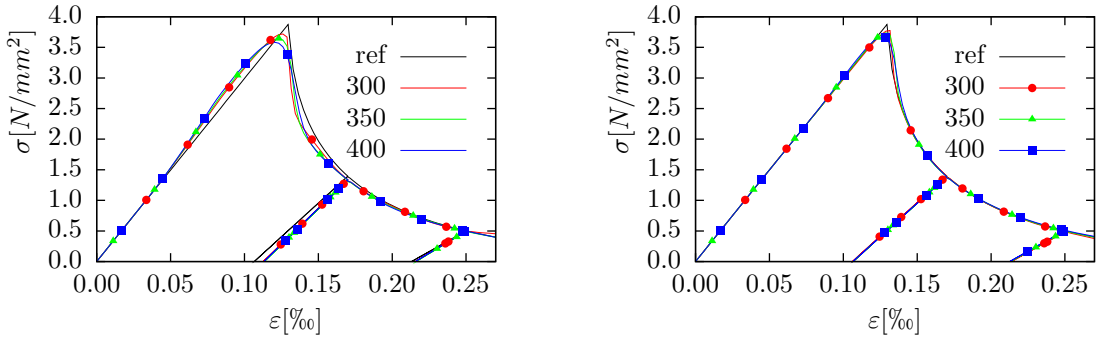
ε^h [‰]	ε^{tot} [‰]	W^{in} $[Nmm]$	σ $[N/mm^2]$	loading/unloading network	stress/strain network
0.0	0.0	0	0	-1	0
0.0	0.2	1.745	0.915	+1	-
0.0	0.4	not calculated		+1	-
0.2	0.0	1.745	-2.774	-1	-2.774
0.2	0.2	1.745	0.915	-1	0.915
0.2	0.4	2.210	0.103	+1	-
0.4	0.0	2.210	-1.723	-1	-1.723
0.4	0.2	2.210	-0.810	-1	-0.810
0.4	0.4	2.210	0.103	-1	0.103

(no unloading) is illustrated in Fig. 8.12, marked as [ref250, ref350,ref450]. It is clearly recognizeable that the dimension of the specimen plays an important role in the stress-strain curve of the homogenized response. This is due to the fact that the size of the localization zone is constant (the weakened element at the center). Consequently, the dissipated energy is almost independent of the length of the model, and by smearing this energy over the length, the sensitivity of the response with respect to the specimen length is obtained.

Using these mesoscale simulations, a neural network (multilayer perceptron) with two hidden layers, each having 6 neurons, is trained with the Levenberg-Marquardt algorithm, until the norm of the gradient is smaller than 10^{-6}. For pure loading, the results are illustrated in Fig. 8.12 for $n_s = 11$ and $n_s = 19$, which corresponds to the number of training samples per dimension. The approximations are almost identical in the pre- and post-peak region. The sharp angle in the reference solution is rounded due to the activation functions with continuous derivatives, but with a higher number of support points the accuracy of the approximation is increased. It is further to be noted that the training data were calculated only for the specimen dimensions 200mm, 300mm, 400mm and 500mm. The reference solutions with specimen dimension 250mm, 350mm and 450mm were not added to the set of training samples. Consequently, the specimen dimension has been interpolated by the neural network.

8.6.2 Macroscale simulation

The macroscopic tension test is performed with a specimen length of 550mm. The dimension of the center element with a weakened cross section to ensure that the localization takes places in this element is varied. Furthermore, loading and unloading are simulated as illustrated in Fig. 8.13. It is to be noted that the unloading stiffness decreases for increasing total strains. This is due to the damage formulation in the mesoscale model. The reference solution is obtained by a pure mesoscale simulation with a mesh size of 50mm. In Fig. 8.13, minor discrepancies for 11 training points compared to the reference solution are visible, whereas with 19 training points there is almost no difference. The first reason is the smoothing within the peak region already illustrated in Fig. 8.12, which is more pronounced for a coarse distribution of training points. The second reason is the approximation of the loading function. Training points on the line $\varepsilon^{tot} = \varepsilon^h$ are per definition in the unloading regime, whereas from a theoretical point of view, this line marks the border between loading and unloading. For



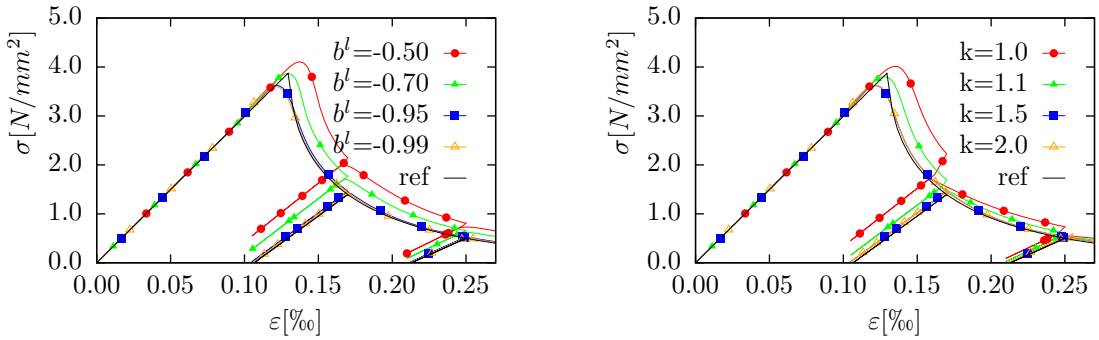
(a) 11 training points (b) 19 training points

Figure 8.13 Load-displacement curve for varying element size with $b^l = -0.99$.

a coarser grid of training points, the distance to the next point in the loading regime is larger and, consequently, the gradient of the loading function is smaller. As a result, the approximated border between loading and unloading is shifted towards the loading regime and therefore the approximated history variable is smaller than the true value.

An important advantage of this approach is that the load-displacement curve is almost independent of the mesh size (ref, 300, 350 and 400 are almost identical), even though localization takes place. A condition for this independency is that the localization takes place within one macroscopic element. Hence, a lower limit of the element size in the macroscale model is given by the size of the localization zone in the mesoscale model. In the example, this size corresponds to the nonlocal influence radius for the damage formulation of 100mm. An upper limit of the element size is only given by the condition that snap back phenomena cannot be modeled with the stress/strain network due to the unique mapping of the strains to stresses.

In Fig. 8.14, the influence of the parameters b^l and k is investigated. The parameter b^l describes the border between loading and unloading in the loading/unloading metamodel. This parameter should be chosen within the range [-1,0], but in order to describe the border



(a) k=2, element size 400 (b) $b^l = 0.95$, element size 400

Figure 8.14 influence of the numerical parameters b^l and k

as accurately a possible, values close to -1 are preferable. On the other hand, it should be assured that all points $\varepsilon^{tot} = \varepsilon^{h}$ are in the unloading regime. This is required to ensure that the modification $\Delta\varepsilon^{h}$ in Eq. (8.8) tends towards the unloading region. Due to numerical reasons it is therefore not recommended to choose a value $f^{b} = -1$, but slightly smaller. In the numerical examples, values in the interval $f^{b}\epsilon[-0.99, -0.95]$ have given accurate results. For higher values a poor description of the border between loading and unloading is obtained and the results differ from the reference solution as illustrated in Fig. 8.14(a).

The second parameter of interest is the parameter k, which ensures that small changes of the total strain only lead to small changes of the history strain. For $k = 1$, which is the lower limit, the change in the norm of the history strain is identical to the change in the norm of the total strain. But $k = 1$ inhibits the transition from one loading zone to another and numerical artifacts arising from the approximative character of the support vector machine describing the loading function cannot be balanced. Consequently, a higher value is preferable, whereas the upper limit is given by numerical reasons, since for $k = \infty$ infinitely small variations of the total strain can lead to large variations of the history strain. Consequently, the response is discontinuous and convergence in the numerical solution might not be obtained. As illustrated in Fig. 8.14(b) a value of $k = 2$ gives accurate results and has been chosen for most of the numerical simulations.

8.7 Summary

In the preceding section, a material model based on neural networks has been introduced. The principal advantage compared to a standard material formulation with a prescribed set of free parameters is the fact that the complexity of the metamodel can be adapted in the course of the training procedure. The presented metamodel distinguishes between loading and unloading. Consequently, a softening in the response function can be approximated as illustrated in the examples. However, a complex history dependent loading regime cannot be represented due to the single history strain used in the formulation. A further limitation of the method is the applicability to low dimensional problems such as e.g. interface formulations, since the number of training data required to obtain an accurate metamodel grows exponentially with the dimension of the input space.

Chapter 9

Conclusions

In this thesis, a framework for the multiscale simulation of concrete has been developed. Starting from a macroscopic model for concrete, the transition was made to a mesoscale model, where, additionally, the stochastic scatter of the response was simulated using random fields. It was further realized that a key problem is the determination of material parameters on the mesoscale, since often only macroscopic tests are available. As a consequence, a parameter identification procedure based on bayesian neural networks has been developed. Furthermore, a macroscopic material model based on a metamodel was introduced, which is build from mesoscale simulations. This allows the application of a mesoscale model to problems on the macroscale while still keeping the computational effort low.

On the macroscopic level, a discrete crack concept in the general framework of the extended finite element method (XFEM) for the simulation of concrete cracking was used. This allowed the discretization of cracks independently of the underlying elements. The cohesive crack concept was used to model the nonlinear relation between crack opening and stresses that can be transfered through the interface, whereas the bulk material is assumed to be linear elastic. Based on this techniques, an automatic adaptive procedure for crack growth simulations has been developed. It allows, on the one hand, for an adaptive refinement of the mesh close to the crack tip using a quadtree data structure, and, on the other hand, for an automatic initiation of new cracks and an extension of existing cracks. Using the automatic refinement procedure, it was possible to start with a relatively coarse mesh and refine only in certain regions which lead to a significant decrease of the computational time.

The key problem was the development of appropriate criteria to decide, whether an existing crack grows and in which direction the crack grows. In order to consider the influence of cohesive forces, a direction criterion based on the minimum of the potential energy was develop and compared to criteria based on linear elastic fracture mechanics (LEFM). It was observed that, for the investigated examples, an approximation of the crack direction based on stress intensity factors from LEFM can give accurate results. Consequently, the cohesive forces can be neglected for the determation of a crack extension, which reduces the numerical effort significantly. The proposed methodology was applied to several standard examples of fracture tests for concrete, and it could be verified that the numerical results accurately reflect the experimental data.

Based on the macroscale model, a mesoscale model for concrete was developed that was able to describe local damage induced by the heterogenous structure of the material. In this model, particles, matrix material and the interfacial transition zone (ITZ) are modeled as separate constituents. Particles, approximated by ellipsoides, were simulated according to a prescribed grading curve and randomly placed into the specimen. It was found that the representation of

the ITZ in the numerical model is crucial for a realistic simulation. Furthermore, localization in uniaxial tests and a stochastic component due to the random particle distribution are naturally included in the model. Originally, it was assumed that the discrete crack concept could be fully transfered to the mesoscale, but due to the interaction between particles and crack tips and the existence of many cracks on the mesoscale, it was realized that a smeared crack concept for the matrix material is advantageous and the discrete crack concept was only used for the ITZ. A material model for the matrix material based on a combined damage-plasticity approach has been developed. The model uses a multisurface plasticity approach with a Drucker-Prager yield surface in compression and a Rankine-criterion in tension. Based on the development of plastic strains formulated in the effective stress space, an isotropic damage formulation was added in order to simulate softening. This required the utilization of a regularization technique in order to reduce the mesh sensitivity of the numerical model. The applied mesh was either an aligned mesh, generated with a common meshing tool and interface elements representing the crack, or a regular mesh, where the ITZ is modeled using the XFEM approach. No significant differences with respect to meshing time and efficiency of the computation could be observed, although the meshing is slightly faster for the regular mesh and the computation is slightly slower due to the additional DOF's. It is assumed that the difference becomes more prominent for 3D calculations. The model was used to simulate a concrete cube under compression. The essential features of the experiment (failure pattern, general shape of the load-displacement curve) are represented in the numerical model.

In an additional part, the stochastic scatter in experimental data is simulated numerically. On the one hand, this scatter can be explained by the intrinsic stochastic distribution of particles in a specimen, and, on the other hand, by spatially variable material parameters represented by random fields. It is demonstrated that the influence of the particle distribution primarily influences the post-peak region. Furthermore, a sensitivity analysis was performed in order to investigate the influence of certain stochastic parameters such as the correlation length on the response. It was found that a significant decrease of macroscopic material parameters (peak load, dissipated energy) is obtained for an intermediate correlation length l_{corr}, whereas the mean responses for the limits of $l_{corr} \to 0$ and $l_{corr} \to \infty$ are almost identical.

In the course of the work it was realized that direct parameter identification procedures for mesoscale models are difficult due to the coupled influence of mesoscale parameters on the macroscopic response. As a consequence, an automatic parameter identification procedure based on bayesian neural networks has been developed. bayesian neural networks are an extension to standard neural networks. The output and free parameters are stochastic variables - and not deterministic as opposed to the standard case. The methodology is based on bayes theorem yielding the most probable distribution of weights under the assumption of a given set of training samples. This procedure inherently incorporates a regularization procedure, thus simplifying the choice of the network architecture, since the problem of overfitting for a network with more neurons than required is resolved. The training algorithm was implemented using a conjugate gradient solver, where it was shown that for bayesian neural networks a preconditioning is mandatory since the condition number of the nonlinear problem gets large due to the influence of the hyperparameters.

A second advantage of bayesian neural networks is the additional information related to the accuracy of the approximation, i.e. how accurate the output can be approximated. In this

thesis, bayesian neural networks have been extended to multiple outputs with a full covariance matrix describing the noise in the approximation. As a consequence, correlations between the output variables can be determined. This procedure has been applied to an inverse problem such as a parameter identification. Material parameters are interpreted as outputs, whereas certain points on the load-displacement curve are used as inputs to the meta-model. It is demonstrated that, with a sufficient number of training samples and input points, the material parameters can be accurately determined. It was shown that it is advantageous to use as many points on the load-displacement curve as possible with the restriction that the numerical model is able to accurately represent the experimental curve. In this context, it is recommended to start with the most complex numerical model available and then, successively, reduce the model to the important parameters. It is further shown that the procedure can be applied to design a set of experiments, which are used to identify material parameters of a numerical model. Additionally, correlations between the identified material parameters are determined, i.e. how is one parameter to be increased, if another is decreased in order to obtain the same response of the numerical model.

A full mesoscale analysis of concrete for realistic dimensions of the structure are beyond the capabilities of the hardware currently available. In order to incorporate the effect of mesoscale models on the macroscopic level, a macroscopic material model based on neural networks is developed. As a consequence, the complexity of the material model is not prescribed, but automatically determined in the training procedure of the network. The model is capable of simulating loading and unloading. This is realized by adding a history variable describing the transition between loading and unloading, which is approximated by a support vector machine. The stress of a material point is then approximated using a neural network. The meta-model is trained using a set of mesoscale simulations. For softening materials, it is realized that the dimension of the mesomodel has a significant influence on the homogenized macroscopic response. Consequently, the dimension of the mesomodel has to be included as additional input parameter of the metamodel. It is further realized that material models based on neural networks are limited to low dimensional spaces, since the number of training samples required to accurately represent the material response grows exponentially with the dimension.

Bibliography

[1] T. Bittencourt, A. R. Ingraffea, J. Llorca. *Fracture Mechanics of Concrete Structures*, chapter Simulation of Arbitrary, Cohesive Crack Propagation, pages 339–350. Elsevier Applied Science, New York, 1992.

[2] I. Carol, M. Lopez, O. Roa. Micromechanical analysis of quasi-brittle materials using fracture-based interface elements. *International Journal for Numerical Methods in Engineering*, 52(1-2):193–215, 2002.

[3] T. Belytschko and T. Black. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering*, 45(5):6001–6020, 1999.

[4] G. Wells and L. Sluys. A new method for modelling cohesive cracks using finite elements. *International Journal for Numerical Methods in Engineering*, 50:2667–2682, 2001.

[5] N. Moës, A. Gravouil, T. Belytschko. Non-planar crack-growth by the extended finite element and level sets - part1: Mechanical model. *International Journal for Numerical Methods in Engineering*, 53:2549–2568, 2002.

[6] F. Erdogan and G. Shih. On the crack extension in plates under plane loading and transverse shear. *Journal of Basic Engineering*, 85:519–527, 1963.

[7] M. Hussain, S. Pu, J. Underwood. Strain energy release rate for a crack under combined mode I and mode II. *Fracture Analysis*, 1974.

[8] G. Sih. Strain-energy-density factor applied to mixed mode crack problems. *International Journal of Fracture*, 10(3):305–321, 1974.

[9] B. Moran and C. Shih. Crack tip and associated domain integrals from momentum and energy balance. *Engineering Fracture Mechanics*, 27(6):615–642, 1987.

[10] G. Nikishkov and S. Atluri. Calculation of fracture mechanics parameters for an arbitrary three-dimensional crack by the 'equaivalent domain integral' method. *International Journal for Numerical Methods in Engineering*, 24(9):1801–1821, 1987.

[11] N. Moës, J. Dolbow, T. Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46(1):131–150, 1999.

[12] N. Moës and T. Belytschko. Extended finite element method for cohesive crack growth. *Engineering Fracture Mechanics*, 69:813–833, 2002.

[13] T. Belytschko, N. Moës, S. Usui, C. Parimi. Arbitrary discontinuities in finite elements. *International Journal for Numerical Methods in Engineering*, 50:993–1013, 2001.

[14] N. Sukumar, Z. Huang, J. Prévost, Z. Suo. Partition of unity enrichment for bimaterial interface cracks. *International Journal for Numerical Methods in Engineering*, 59:1075–1102, 2004.

[15] N. Sukumar, D. Chopp, N. Moës, T. Belytschko. Modelling holes and inclusions by level sets in the extended finite-element method. *Computation Methods in Applied Mechanics and Engineering*, 190:6183–6200, 2001.

[16] N. Moës, M. Cloirec, P. Cartraud, J.-F. Remacle. A computational approach to handle complex microstructure geometries. *Computation Methods in Applied Mechanics and Engineering*, 192:3163–3177, 2003.

[17] T. Hettich and E. Ramm. Interface material failure modeled by the extended finite-element method and level sets. *Computer Methods in Applied Mechanics and Engineering*, accepted for publication, 2006.

[18] J. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Science*, 93(4):1591–1595, 1996.

[19] D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang. A pde-based fast local level set method. *Journal of Computational Physics*, 155:410–438, 1999.

[20] M. Sussman, P. Smereka, S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114(1):146–159, 1994.

[21] M. Kaplan. Crack propagation and the fracture of concrete. *ACI Journal*, 58(11):591–610, 1961.

[22] C. Kesler, D. Naus, J. Lott. Fracture mechanics - its applicability to concrete. In *Proceedings of the International conference on the mechanical behavior of materials, 1971*, volume 4, pages 113–124, Kyoto, 1972. The society of materials science.

[23] A. Hillerborg, M. Modéer, P. Petersson. Analysis of crack formation and crack growth in concrete by means of fracture mechanics and finite elements. *Cement and Concrete Research*, 6:773–782, 1976.

[24] D. Dugdale. Yielding of sheets containing slits. *Journal of Mechanics and Physics of Solids*, 8:100–104, 1960.

[25] G. Barenblatt. The mathematical theory of equilibrium of cracks in brittle fracture. *Advances in Applied Mechanics*, 7:55–129, 1962.

[26] P. Petersson. Crack growth and development of fracture process zone in plain concrete and similar materials. Report TVBM-100, Division of Building Materials, Lund Institute of Technology, Lund, Sweden, 1981.

[27] R. Rots. *Softening of concrete loaded in compression.* PhD thesis, TU Eindhoven, 1992.

[28] J. Červenka. *Discrete crack modelling in concrete structures.* PhD thesis, University of Colorado, Boulder, 1994.

[29] I. Carol, P. Prat, López. Normal/shear cracking model: Application to discrete crack analysis. *Journal of Engineering Mechanics, ASCE*, 132:765–773, 1997.

[30] V. Tvergaard. Cohesive zone representation of failure between elastic or rigid solids and ductile solids. *Engineering Fracture Mechanics*, 70:1859–1868, 2003.

[31] M. Ortiz and A. Pandolfi. Finite-deformation irreversible cohesive elements for three-dimensional crack-propagation analysis. *International Journal for Numerical Methods in Engineering*, 44(9):1267–1282, 1999.

[32] M. Xie. *Finite element modelling of discrete crack propagation*. PhD thesis, University of New Mexico, 1995.

[33] Z. Yang. An energy-based crack growth criterion for modelling elastic-plastic ductile fracture. *Mechanics Research Communications*, 32:514–524, 2004.

[34] J. Oliver and A. Huespe. Continuum approach to material failure in strong discontinuity settings. *Computer Methods in Applied Mechanics and Engineering*, 193(30-32):3195–3220, 2004.

[35] D. Cendón, J. Gálvez, M. Elices, J. Planas. Modelling the fracture of concrete under mixed loading. *International Journal of Fracture*, 103(3):293–310, 2000.

[36] T. Nishioka. A numerical study of the use of path independent integrals in elastic-dynamic crack propagation. *Engineering Fracture Mechanics*, 18:23–33, 1983.

[37] P. Dumstorff and G. Meschke. Investigation of crack growth criteria in the context of the extended finite element method. In P. Neittaanmäki *et al.*, editors, *Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering*, 2004.

[38] T. Most. *Stochastic crack growth simulation in reinforced concrete structures by means of coupled finite element and meshless methods*. PhD thesis, Bauhaus University Weimar, 2005.

[39] H. Samet. *The design and analysis of spatial data structures*. Addison-Wesley, Reading, MA, 1990.

[40] M. Jirásek and B. Patzák. Adaptive resolution of localized damage in quasibrittle materials. *Journal of Engineering Mechanics, ASCE*, 130(6):720–732, 2003.

[41] N. Sukumar. Construction of polygonal interpolants: a maximum entropy approach. *International Journal for Numerical Methods in Engineering*, 61:2159–2181, 2004.

[42] F. Stazi, E. Budyn, T. Belytschko. An extended finite element method with higher-order elements for curved cracks. *Computation Mechanics*, 31:38–48, 2003.

[43] J. Chessa, H. Wang, T. Belytschko. On the construction of blending elements for local partition of unity enriched finite elements. *International Journal for Numerical Methods in Engineering*, 57(7):1015–1038, june 2003.

[44] J. Dolbow, N. Moës, T. Belytschko. Discontinuous enrichment in finite elements with a partition of unity method. *Finite Elements in Analysis and Design*, 36:235–260, 2000.

[45] J. R. Shewchuk. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In M. C. Lin and D. Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag, May 1996. From the First ACM Workshop on Applied Computational Geometry.

[46] J. C. J. Schellekens and R. de Borst. On the numerical integrations of interface elements. *International Journal for Numerical Methods in Engineering*, 36(1):43–66, 1993.

[47] P. Amestoy, I. Duff, J.-Y. L'Excellent. Multifrontal parallel distributed symmetric and unsymmetric solvers. *ENSEEIHT-IRIT Technical Report, Revised version appeared in Comput. Methods in Appl. Mech. Eng.*, 184:501–520, 2000.

[48] A. Goldstein. On steepest descent. *Journal of the Society for Industrial and Applied Mathematics, Series A: Control*, 3(1):147–151, 1965.

[49] L. Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1):1–3, 1966.

[50] J. E. Dennis and J. J. Moré. Quasi-newton methods, motivation and theory. *SIAM Review*, 19:46–89, 1977.

[51] The MathWorks, Inc. *MATLAB: Partial Differential Equation Toolbox, User's Guide, release 2008a*, 2008.

[52] A. Unterkircher and J. Reissner. Parallel assembling and equation solving via graph algorithms with an application to the FE simulation of metal extrusion processes. *Computers & Structures*, 83(8-9):627–638, 2005.

[53] B. Winkler. *Traglastuntersuchungen von unbewehrten und bewehrten Betonstrukturen auf der Grundlage eines objektiven Werkstoffgesetzes für Beton.* PhD thesis, University of Innsbruck, 2001.

[54] M. Nooru-Mohamed. *Mixed-mode fracture of concrete: An experimental approach.* PhD thesis, Delft University of Technology, Delft, 1992.

[55] C. Feist, W. Kerber, H. Lehar, G. Hofstetter. A comparative study of numerical models for concrete cracking. In P. Neittaanmäki *et al.*, editors, *Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS*, 2004.

[56] K. J. Willam and E. P. Warnke. Constitutive model for the triaxial behavior of concrete. *Proceedings, International Association for Bridge and Structural Engineering*, 19:1–30, 1975.

[57] N. S. Ottosen. A failure criterion for concrete. *ASCE Journal of Engineering Mechanics Division*, 103(4):527–535, 1977.

[58] S. S. Hsieh, E. C. Ting, W. F. Chen. Plastic-fracture model for concrete. *International Journal of Solids and Structures*, 18(3):181–197, 1982.

[59] W.-F. Chen. *Plasticity in reinforced concrete.* McGraw-Hill, New York, 1982.

[60] P. M. Naghdi and J. A. Trapp. Significance of formulating plasticity theory with reference to loading surfaces in strain space. *International Journal of Engineering Science*, 13(9-10):785–797, 1975.

[61] L. Kachanov. О времени разрушения в условиях ползучести(time of the rupture process under creep conditions). *Известия Академии наук СССР. Отделение технических наук.*, 8:26–31, 1958.

[62] L. Kachanov. *Introduction to Continuum Damage Mechanics*. Kluwer Academic, 1986.

[63] J. Janson and J. Hult. Fracture mechanics and damage mechanics—a combined approach. *Journal de M'ecanique Applique'e*, 1:69–84, 1977.

[64] J. Mazars and G. Pijaudier-Cabot. From damage to fracture mechanics and conversely: A combined approach. *International Journal of Solids and Structures*, 33(20-22):3327–3342, 1996.

[65] J. Mazars. A description of micro- and macroscale damage of concrete structures. *Engineering Fracture Mechanics*, 25(5-6):729–737, 1986.

[66] J. P. Cordebois and F. Sidoroff. Anisotropic damage in elasticity and plasticity. *Journal de mecanique theorique et appliquee*, pages 45–59, 1980.

[67] J. L. Chaboche. Continuous damage mechanics - a tool to describe phenomena before crack initiation. *Nuclear Engineering and Design*, 64(2):233–247, 1981.

[68] C. L. Chow and J. Wang. An anisotropic theory of elasticity for continuum damage mechanics. *International Journal of Fracture*, 33(1):3–16, 1987.

[69] Y. Rashid. Analysis of prestressed concrete pressure vessels. *Nuclear Engineering and Design*, 7:334–344, 1968.

[70] Z. P. Bažant and B. H. Oh. Crack band theory for fracture of concrete. *Matériaux et constructions*, 16(93):155–177, 1983.

[71] R. Litton. *A contribution to the analysis of concrete structures under cyclic loading*. PhD thesis, university of California, Berkley, 1974.

[72] A. K. Gupta and H. Akbar. Cracking in reinforced concrete analysis. *Journal of Structural Engineering*, 110(8):1735–1746, 1984.

[73] K. Willam, E. Pramono, S. Sture. Fundamental issues of smeared crack models. In S. Shah and S. Swartz, editors, *Proceedings of SEM/RILEM International Conference on Fracture of Concrete and Rock*, pages 192–207, 1987.

[74] M. A. Crisfield and J. Wills. Analysis of r/c panels using different concrete models. *Journal of Engineering Mechanics*, 115(3):578–597, 1989.

[75] J. G. Rots and J. Blaauwendraad. Crack models for concrete: discrete or smeared? fixed, multi-directional or rotating? *Heron*, 34(1):3–59, 1989.

[76] J. Rots. *Computional modeling of concrete fracture*. PhD thesis, Delft university of technology, 1988.

[77] G. I. Taylor. Plastic strain in metals. *Journal of the Institute of Metals, London*, 62:307–324, 1938.

[78] S. B. Batdorf and B. Budiansky. A mathematical theory of plasticity based on the concept of slip. Technical report, Tech Note Note. 1871, National Advisory Committee for Aeronautics, Washington, D.C., 1949.

[79] Z. P. Bažant and P. G. Gambarova. Crack shear in concrete: crack band microplane model. *Journal of Structural Engineering*, 110(9):2015–2035, 1984.

[80] Z. P. Bažant and B. H. Oh. Microplane model for progressive fracture of concrete and rock. *Journal of Engineering Mechanics*, 111(4):559–582, 1985.

[81] Z. P. Bažant and P. C. Prat. Microplane model for brittle-plastic material: I. theory. *Journal of Engineering Mechanics*, 114(10):1672–1688, 1988.

[82] Z. P. Bažant, F. C. Caner, I. Carol, M. D. Adley, S. A. Akers. Microplane model M4 for concrete I: Formulation with work-conjugate deviatoric stress. *Journal of Engineering Mechanics*, 126(9):944–953, 2000.

[83] J. Oliver, M. Cervera, S. Oller, J. Lubliner. Isotropic damage models and smeared crack analysis of concrete. In N. Bićanić and H. Mang, editors, *Computer Aided Analysis and Design of Concrete Structures*, pages 945–957. Pineridge Press, Swansea, 1990.

[84] Z. P. Bažant. Instability, ductility, and size effect in strain-softening concrete. *ASCE Journal of Engineering Mechanics Division*, 102(2):331–344, 1976.

[85] D. G. B. Edelen, A. E. Green, , N. Laws. Nonlocal continuum mechanics. *Archive for Rational Mechanics and Analysis*, 43(1):36–44, 1971.

[86] A. C. Eringen. Linear theory of nonlocal elasticity and dispersion of plane waves. *International Journal of Engineering Science*, 10(5):425–435, 1972.

[87] A. C. Eringen. On nonlocal plasticity. *International Journal of Engineering Science*, 19(12):1461–1474, 1981.

[88] Z. P. Bažant and T.-P. Chang. Instability of nonlocal continuum and strain averaging. *Journal of Engineering Mechanics*, 110(10):1441–1450, 1984.

[89] Z. P. Bažant and G. Pijaudier-Cabot. Nonlocal continuum damage, localization instability and convergence. *Journal of Applied Mechanics, Transactions ASME*, 55(2):287–293, 1988. Cited By (since 1996): 192.

[90] C. Saouridis and J. Mazars. Prediction of the failure and size effect in concrete via bi-scale damage approach. *Engineering computations*, 9(3):329–344, 1992.

[91] Z. P. Bažant and M. Jirásek. Nonlocal integral formulations of plasticity and damage: Survey of progress. *Journal of Engineering Mechanics*, 128(11):1119–1149, 2002.

[92] E. C. Aifantis. On the microstructural origin of certain inelastic models. *Journal of Engineering Materials and Technology, Transactions of the ASME*, 106(4):326–330, 1984.

[93] N. Triantafyllidis and E. C. Aifantis. A gradient approach to localization of deformation. I. hyperelastic materials. *Journal of Elasticity*, 16(3):225–237, 1986.

[94] H.-B. Mühlhaus and E. C. Alfantis. A variational principle for gradient plasticity. *International Journal of Solids and Structures*, 28(7):845–857, 1991.

[95] M. Jirásek. Nonlocal models for damage and fracture: Comparison of approaches. *International Journal of Solids and Structures*, 35(31-32):4133–4145, 1998.

[96] I. Carol, E. Rizzi, K. Willam. On the formulation of anisotropic elastic degradation : II. generalized pseudo-rankine model for tensile damage. *International Journal of Solids and Structures*, 38(4):519–546, 2001.

[97] E. Hansen, K. Willam, I. Carol. A two-surface anisotropic damage/plasticity model for plain concrete. In R. de Borst, J. Mazars, G. Pijaudier-Cabot, J. van Mier, editors, *Fracture mechanics of concrete structures*, pages 549–556, 2001.

[98] J. Ju. On energy-based coupled elastoplastic damage theories: Constitutive modeling and computational aspects. *International Journal of Solids and Structures*, 25(7):803–833, 1989.

[99] L. Jason, A. Huerta, G. Pijaudier-Cabot, S. Ghavamian. An elastic plastic damage formulation for concrete: application to elementary tests and comparison with an isotropic damage model. *Computer Methods in Applied Mechanics and Engineering*, 195(52):7077–7092, 2006.

[100] J. Lubliner, J. Oliver, S. Oller, E. Oñate. A plastic damage model for concrete. *International Journal of Solids and Structures*, 25(3):299–326, 1989.

[101] S. Ananiev and J. Ožbolt. Plastic damage model for concrete in principal directions. In V. Li, C. Leung, K. Willam, S. Billington, editors, *Fracture mechanics of concrete structures*, pages 271–278, 2004.

[102] P. Grassl and M. Jirásek. Damage-plastic model for concrete failure. *International Journal of Solids and Structures*, 43(22-23):7166–7196, 2004.

[103] P. Grassl and M. Jirásek. Plastic model with non-local damage applied to concrete. *International Journal for Numerical and Analytical Methods in Geomechanics*, 30(1):71–90, 2006.

[104] W. T. Koiter. Stress-strain relations, uniqueness and variational theromes for elastic-plastic materials with a singular yield surface. *Quaterly of applied mathematics*, 11:350–354, 1953.

[105] J. Mandel. Une généralisation de la théorie de la plasticité de W.T. Koiter. *International Journal of Solids and Structures*, 1:273–295, 1965.

[106] J. Simo and T. Hughes. *Computational Inelasticity*. Springer-Verlag, 1997.

[107] P. Pivonka. *Constitutive modeling of triaxiality loaded concrete considering large compressive stresses: application to pullout tests of anchor bolts*. PhD thesis, Technical university Vienna, 2001.

[108] Z. P. Bažant and G. Pijaudier-Cabot. Measurement of characteristic length of nonlocal continuum. *Journal of Engineering Mechanics*, 115(4):755–767, 1989.

[109] P. Grassl and M. Jirásek. Meso-mechanically motivated nonlocal models for modelling of the fracture process zone in quasi-brittle materials. In *8th. World Congress on Computational Mechanics (WCCM8) 5th European Congress on Computational Methods in Applied Sciences and Engineeering (ECCOMAS 2008)*, 2008.

[110] S. Häfner, S. Eckardt, T. Luther, C. Könke. Mesoscale modeling of concrete: Geometry and numerics. *Computers & Structures*, 84(7):450–461, 2006.

[111] A. Carpinteri and B. Chiaia. Multifractal nature of concrete fracture surfaces and size effects on nominal fracture energy. *Matériaux et constructions*, 28(182):435–443, 1995.

[112] G. Nagai, T. Yamada, A. Wada. Three-dimensional nonlinear finite element analysis of the macroscopic compressive failure of concrete materials based on real digital image. *Computing in Civil and Building Engineering*, 1:449–456, 2000.

[113] E. Garboci. Three-dimensional mathematical analysis of particle shape using X-ray tomography and spherical harmonics: Application to aggregates used in concrete. *Cement and Concrete Research*, 32:1621–1638, 2002.

[114] E. Garboci and J. Bullard. Shape analysis of a reference cement. *Cement and Concrete Research*, 34:1933–1937, 2004.

[115] Z. Wang, A. Kwan, H. Chan. Mesoscopic study of concrete i: Generation of random aggregate structure and finite element mesh. *Computers & Structures*, 70(5):533–544, 1999.

[116] E. Schlangen. *Experimental and Numerical Analysis of Fracture Processes in Concrete*. PhD thesis, Delft University of technology, 1993.

[117] A. Vervuurt. *Interface Fracture in Concrete*. PhD thesis, Delft University of technology, 1997.

[118] B. Chiaia, A. Vervuurt, J. Van Mier. Lattice model evaluation of progressive failure in disordered particle composites. *Engineering Fracture Mechanics*, 57(2-3):301–318, 1997.

[119] J. Leite, V. Slowik, H. Mihashi. Computer simulation of fracture processes of concrete using mesolevel models of lattice structures. *Cement and Concrete Research*, 34(6):1025–1033, 2004.

[120] P. Grassl, Z. P. Bažant, G. Cusatis. Lattice-cell approach to quasibrittle fracture modeling. In N. Bianic, R. de Borst, H. Mang, G. Meschke, editors, *Proceedings Intern. Conf. on Computational Modelling of Concrete Structures (EURO-C 2006)*, 2006.

[121] G. A. d'Addetta. *Discrete models for cohesive fricional materials*. PhD thesis, University of Stuttgart, Institute for Structural Mechanics, 2002.

[122] P. Wriggers and S. O. Moftah. Mesoscale models for concrete: Homogenisation and damage behaviour. *Finite Elements in Analysis and Design*, 42(7 SPEC. ISS.):623–636, 2006.

[123] S. Eckardt and C. Könke. Simulation of damage in concrete structures using multiscale methods. In N. Bianic, R. de Borst, H. Mang, G. Meschke, editors, *Proceedings Intern. Conf. on Computational Modelling of Concrete Structures (EURO-C 2006)*, 2006.

[124] J. F. Unger and C. Könke. Simulation of concrete using the extended finite element method. In N. Bićanić, R. de Borst, H. Mang, G. Meschke, editors, *Proceedings Intern. Conf. on Computational Modelling of Concrete Structures (EURO-C 2006)*, pages 239–247. Balkema, 2006.

[125] K. L. Scrivener. Backscattered electron imaging of cementitious microstructures: understanding and quantification. *Cement and Concrete composites*, 26:935–945, 2004.

[126] W. A. Tasong, C. J. Lynsdale, J. C. Cripps. Aggegate-cement paste interface II: Influence of aggregate physical properties. *Cement and Concrete Research*, 28(10):1453–1465, 1998.

[127] K. M. Lee, O. Buyukozturk, A. Oumera. Fracture analysis of mortar-aggregate interfaces in concrete. *Journal of engineering mechanics*, 118(10):2031–2047, 1992.

[128] S. Caliskan, B. Karihaloo, B. Barr. Study of rock-mortar interfaces. part I: surface roughness of rock aggregates and microstructural characteristics of interface. *Magazine of Concrete Research*, 54(6):449–461, 2002.

[129] K. Walz. Festigkeitsentwicklung von Beton im Alter von 30 bis 50 Jahren. *Beton*, 3:95–98, 1976.

[130] K. Walz. Festigkeitsentwicklung von Beton im Alter von 30 bis 50 Jahren. *Beton*, 4:135–138, 1976.

[131] K.-Y. Liao, P.-K. Chang, P. Yaw-Nan, C.-C. Yang. A study on characteristics of interfacial transition zone in concrete. *Cement and Concrete Research*, 34:977–989, 2004.

[132] S. Diamond and J. Huang. The ITZ in concrete - a different view based on image analysis and SEM observations. *Cement & Concrete Composites*, 23:179–188, 2001.

[133] W. A. Tasong, J. C. Lynsdale, J. C. Cripps. Aggegate-cement paste interface i: Influence of aggregate geochemistry. *Cement and Concrete Research*, 29:1019–1025, 1999.

[134] G. Appa Rao and B. Raghu Prasad. Influence of the roughness of aggregate surface on the interface bond strength. *Cement and Concrete Research*, 32:253–257, 2002.

[135] S. Caliskan, B. Karihaloo, B. Barr. Study of rock-mortar interfaces. part II: Strength of interface. *Magazine of Concrete Research*, 54(6):463–472, 2002.

[136] M. J. Aquino, Z. Li, S. P. Shah. Mechanical properties of the aggregate and cement interface. *Advanced Cement Based Materials*, 2:211–223, 1995.

[137] G. Guinea, K. El-Sayed, C. Rocco, M. Elices, J. Planas. The effect of the bond between the matrix and the aggregates on the cracking mechancism and fracture parameters of concrete. *Cement and Concrete Research*, 32:1961–1970, 2002.

[138] L. A. Feldkamp, S. A. Goldstein, A. M. Parfitt, G. Jesion, M. Kleerekoper. The direct examination of three-dimensional bone architecture in vitro by computed tomography. *Journal of Bone and Mineral Research*, 4(1):3–11, 1989.

[139] N. Takano, K. Kimura, M. Zako, F. Kubo. Multi-scale analysis and microscopic stress evaluation for ceramics considering the random microstructures. *JSME International Journal, Series A: Solid Mechanics and Material Engineering*, 46(4):527–535, 2003.

[140] S. J. Hollister and N. Kikuchi. Homogenization theory and digital imaging: A basis for studying the mechanics and design principles of bone tissue. *Biotechnology and Bioengineering*, 43(7):586–596, 1994.

[141] Y. Nagano, Y. Ikeda, H. Kawamoto. Application of 3D X-RAY CT to stress simulation analysis of porous materials with homogenization method. In C. Miyasaka, Y. Yokono, D. Bray, Cho, editors, *ASME/JSME Pressure Vessels and Piping Conference, San Diego*, volume 484, pages 141–146, 2004.

[142] Y. B. Zaitsev and F. H. Wittmann. Simulation of crack propagation and failure of concrete. *Materials and Structures*, 14(5):357–365, 1981.

[143] Z. P. Bažant, M. R. Tabbara, M. T. Kazemi, G. Pijaudier-Cabot. Random particle model for fracture of aggregate or fiber composites. *Journal of Engineering Mechanics*, 116(8):1686–1705, 1990.

[144] E. Schlangen and J. G. M. van Mier. Simple lattice model for numerical simulation of fracture of concrete materials and structures. *Materials and Structures*, 25(9):534–542, 1992.

[145] S. Häfner, S. Eckardt, C. Könke. A geometrical inclusion-matrix model for the finite element analysis of concrete at multiple scales. In *Proceedings of the 16th International Conference on the Application of Computer Science and Mathematics in Architecture and Civil Engineering*, Weimar, 2003.

[146] J. P. B. Leite, V. Slowik, J. Apel. Computational model of mesoscopic structure of concrete for simulation of fracture processes. *Computers & Structures*, 85(17-18):1293–1303, 2007.

[147] S. Eckardt and C. Könke. Adaptive simulation of the damage behavior of concrete using heterogeneous multiscale methods. In K. Gürlebeck and C. Könke, editors, *Proceedings of the 17th International Conference on the Application of Computer Science and Mathematics in Architecture and Civil Engineering*, Weimar, 2006.

[148] W. Wang, W. Jiaye, K. Myung-Soo. An algebraic condition for the separation of two ellipsoids. *Computer aided Geometric Design*, 18(6):531–539, 2001.

[149] D. Hook and P. McAree. *Graphics gems*, chapter Using Sturm sequences to bracket real roots of polynomial equations, pages 416–422. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[150] F. H. Wittmann, P. E. Roelfstra, H. Sadouki. Simulation and analysis of composite structures. *Materials Science and Engineering*, 68(2):239–248, 1985.

[151] S. Eckardt. *Adaptive heterogeneous multiscale models for the nonlinear simulation of concrete*. PhD thesis, Bauhaus-University Weimar, 2009.

[152] J. R. Shewchuk. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In M. C. Lin and D. Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag, 1996.

[153] C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities, october 2008. `http://geuz.org/gmsh/`.

[154] M. R. A. van Vliet and J. G. M. van Mier. Softening behavior of concrete under uniaxial compression. In F. Wittman, editor, *Fracture mechanics of concrete structures (FraMCoS-2, Zürich)*, pages 383–396, 1995.

[155] A. Reuss. Berechnung der Fließgrenze von Mischkristallen auf Grund der Plastizitätsbedingung für Einkristalle. *Zeitschrift für Angewandte Mathematik und Mechanik*, 9(1):49–58, 1929.

[156] B. E. Barrágan. *Failure and toughness of steel fibre reinforced concrete under tension and shear*. PhD thesis, Universitat politècnica de Catalunya, 2002.

[157] B. Möller, W. Graf, M. Beer. Safety assessment of structures in view of fuzzy randomness. *Computers & Structures*, 81:1567–1582, 2003.

[158] *Fuzzy Sets and Fuzzy Logic: The foundations of application from a Mathematical Point of View*. Vieweg, Wiesbaden, 1993.

[159] H.-J. Zimmermann. *Fuzzy Set Theory and its Applications*. Kluwer Academic Publishers, 2001.

[160] W. Dong and H. C. Shah. Vertex method for computing functions of fuzzy variables. *Fuzzy Sets Syst.*, 24(1):65–78, 1987.

[161] B. Möller, W. Graf, M. Beer. Fuzzy structural analysis using α-level optimization. *Computational Mechanics*, 26(6):547–565, 2000.

[162] B. Möller and M. Beer. *Fuzzy Randomness*. Springer, Heidelberg, 2004.

[163] A. Der Kiureghian. Analysis of structural reliability under parameter uncertainties. *Probabilistic Engineering Mechanics*, 23(4):351–358, 2008.

[164] M. Matsumoto and T. Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.*, 8(1):3–30, 1998.

[165] D. J. Mackay. *Information theory, inference and learning algorithms*. Cambridge University Press, 2005.

[166] L. Carlitz. The inverse of the error function. *Pacific Journal of Mathematics*, 13(2):459–470, 1963.

[167] G. Marsaglia and W. W. Tsang. The ziggurat method for generating random variables. *Journal of Statistical Software*, 5(8):1–7, 2000.

[168] C. C. S. Wallace. Fast pseudorandom generators for normal and exponential variates. *ACM Transactions on Mathematical Software*, 22(1):119–127, 1996.

[169] G. E. P. Box and M. E. Muller. A note on the generation of random normal deviates. *The Annals of Mathematical Statistics*, 29(2):610–611, 1958.

[170] L. Devroye. *Non-Uniform Random Variate Generation*. Springer-Verlag, New York, 1986.

[171] D. B. Thomas, W. Luk, P. H. Leong, J. D. Villasenor. Gaussian random number generators. *ACM Computing Survey*, 39(4):11, 2007.

[172] C. Spearman. "general intelligence" objectively determined and measured. *American Journal of Psychology*, 15:201–293, 1904.

[173] M. Kendall. A new measure of rank correlation. *Biometrika*, 30:81–89, 1938.

[174] P.-L. Liu and A. Der Kiureghian. Multivariate distribution models with prescribed marginals and covariances. *Probabilistic Engineering Mechanics*, 1(2):105–112, 1986.

[175] A. Nataf. Détermination des distribution de probabilités dont les marges sont données. *Compte Rendus de l'Académie des Sciences*, 225(2):42–43, 1962.

[176] M. Mackay, R. Beckman, W. Conover. A comparison of three methods for selecting values of output variables in the analysis of output from a computer code. *Technometrics*, 21:239–245, 1979.

[177] E. Saliby. Descriptive sampling. a better approach to monte carlo simulation. *Journal of the Operational Research Society*, 41(12):1133–1142, 1990.

[178] E. Saliby. Descriptive sampling: An improvement over latin hypercube sampling. In *Winter Simulation Conference, Proceedings of the 29th conference on Winter simulation*, pages 230–233, 1997.

[179] D. E. Huntington and C. S. Lyrintzis. Improvements to and limitations of latin hypercube sampling. *Probabilistic Engineering Mechanics*, 13(4):245–253, 1998.

[180] R. Iman and W. Conover. A distribution-free approach to inducing rank correlation among input variables. *Communications in Statistics - Simulation and Computation*, 11(3):311–334, 1982.

[181] A. Florian. An efficient sampling scheme: Updated latin hypercube sampling. *Probabilistic Engineering Mechanics*, 7(2):123–130, 1992.

[182] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[183] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[184] C. Brenner. *Ein Beitrag zur Zuverlässigkeitsanalyse von Strukturen unter Berücksichtigung von Systemunsicherheiten mit Hilfe der Methode der stochastischen finiten Elemente*. PhD thesis, Bauhaus university, Weimar, 1995.

[185] E. H. Vanmarcke. Probabilistic modelling of soil profiles. *ASCE Journal of the Geotechnical Engineering Division*, 103(11):1227–1246, 1977.

[186] E. H. Vanmarcke. *Random fields: Analysis and Synthesis*. The MIT Press, Cambridge, Massachusetts, 1983.

[187] R.-H. Cherng and Y. K. Wen. Reliability of uncertain nonlinear trusses under random excitation. i. *Journal of Engineering Mechanics*, 120(4):733–747, 1994.

[188] L. W. Kam, T. Belytschko, A. Mani. Random field finite elements. *International Journal for Numerical Methods in Engineering*, 23(10):1831–1845, 1986.

[189] G. Deodatis. Simulation of ergodic multivariate stochastic processes. *Journal of Engineering Mechanics*, 122(8):778–787, 1996.

[190] R. Popescu. *Stochastic variability of soil properties: data analysis, digital simulation, effects on system behavior*. PhD thesis, Princeton unversity, 1995.

[191] E. Vanmarcke and M. Grigoriu. Stochastic finite element analysis of simple beams. *Journal of Engineering Mechanics*, 109(5):1203–1214, 1983.

[192] T. Soong and M. Grigoriu. *Random Vibration of Mechanical and Structural Systems*. Prentice Hall, 1992.

[193] M. Shinozuka and G. Deodatis. Simulation of multi-dimensional gaussian stochastic fields by spectral representation. *Applied Mechanics Reviews*, 49(1):29–53, 1996.

[194] T. Oooura. General purpose fft (fast fourier/cosine/sine transform) package, september 2008.
`http://www.kurims.kyoto-u.ac.jp/~ooura/fft.html`.

[195] L. A. Bergman *et al.* A state-of-the-art report on computational stochastic mechanics. *Probabilistic Engineering Mechanics*, 12(4):197–321, 1997.

[196] H. G. Matthies, C. E. Brenner, C. G. Bucher, C. G. Soares. Uncertainties in probabilistic numerical analysis of structures and solids - stochastic finite elements. *Structural Safety*, 19(3):283–336, 1997.

[197] D. Novák. *Computational structures technology*, chapter Stochastic modelling of failure and size effect of concrete structures, pages 93–122. Civil-Comp press, Edinburgh, UK, UK, 2002.

[198] M. Vořechovský. Interplay of size effects in concrete specimens under tension studied via computational stochastic fracture mechanics. *International Journal of Solids and Structures*, 44(9):2715 – 2731, 2007.

[199] T. Most and C. Bucher. Probabilistic analysis of concrete cracking using neural networks and random fields. *Probabilistic Engineering Mechanics*, 22(2):219 – 229, 2007.

[200] Z. Yang and X. F. Xu. A heterogeneous cohesive model for quasi-brittle materials considering spatially varying random fracture properties. *Computer Methods in Applied Mechanics and Engineering*, 197(45-48):4027–4039, 2008.

[201] M. Vořechovský. Simulation of simply cross correlated random fields by series expansion methods. *Structural Safety*, 30(4):337–363, 2008.

[202] C. Kessler-Kramer. *Zugtragverhalten von Beton unter Ermüdungsbeanspruchung*. PhD thesis, Universität Karlsruhe (TH), Germany, 2002.

[203] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, 5:115–133, 1943.

[204] D. O. Hebb. *The Organization of Behavior: a neuropsychological approach*. Wiley, New York, 1949.

[205] F. Rosenblatt. *Principles of Neurodynamics*. Spartan Books, New York, 1962.

[206] M. Misky and S. P. Papert. *Perceptrons*. MIT Press, 1969.

[207] J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558, 1982.

[208] B. Kosko. Bidirection associative memories. *IEEE Transactions on Systems, Man and Cybernetics*, 18(1):49–60, 1988.

[209] D. Rumelhart and J. McClelland, editors. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition Vol 1: Foundations*. MIT Press, 1986.

[210] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[211] K. Hornik, M. Stinchcombe, H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[212] D. E. Rumelhart, G. E. Hinton, R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

[213] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *Proc. of the IEEE Intl. Conf. on Neural Networks*, pages 586–591, San Francisco, CA, 1993.

[214] E. M. Johansson, F. U. Dowla, D. M. Goodman. Backpropagation learning for multi-layer feed-forward neural networks using the conjugate gradient method. *International Journal of Neural Systems*, 2(4):291–301, 1992.

[215] M. F. Moller. Scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6(4):525–533, 1993.

[216] M. H. Hagan and M. B. Menhaj. Training feedforward networks with the marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6):989–993, 1994.

[217] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.

[218] R. Fletcher. *Practical Methods of Optimization*. Wiley, 1987.

[219] R. Fletcher and C. M. Reeves. Function minimization by conjugate gradients. *Computing Journal*, 7(2):149–154, 1964.

[220] E. Polak and G. Ribière. Note sur la convergence de directions conjugées. *Revue Francaise d'Informatique et de Recherche Opérationelle*, 16:35–43, 1969.

[221] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*, chapter Minimization or Maximization of Functions, pages 402–405. Cambridge University Press, 1988.

[222] M. J. D. Powell. Restart procedures for the conjugate gradient method. *Mathematical Programming*, 12(1):241–254, 1977.

[223] R. Brent. *Algorithms for Minimization without Derivatives*. Englewood Cliffs, NJ: Prentice-Hall, 1973.

[224] S. Haykin. *Neural Networks, a comprehensive foundation*, chapter 4.6, pages 181–182. Prentice Hall, second edition, 1999.

[225] M. Hernandez-Espinosa, C.; Fernandez-Redondo. Multilayer feedforward weight initialization. In *1. International Joint Conference on Neural Networks*, pages 166 – 170, 2001.

[226] Y. LeCun, J. S. Denker, S. Solla, R. E. Howard, L. D. Jackel. Optimal brain damage. In D. Touretzky, editor, *Advances in Neural Information Processing Systems 2 (NIPS*89)*, Denver, CO, 1990. Morgan Kaufman.

[227] B. Hassibi, D. G. Stork, G. J. Wolff. Optimal brain surgeon and general network pruning. In *IEEE International Conference on Neural Networks*, volume 1, pages 293–299, 1993.

[228] R. Reed. Pruning algorithms - a survey. *IEEE Transactions on Neural Networks*, 4(5):740–747, 1993.

[229] S. E. Fahlman and C. Lebiere. The cascade-correlation learning architecture. In *Advances in Neural Information Processing Systems 2*, pages 524–532. Morgan Kaufmann, 1990.

[230] A. Krogh and J. A. Hertz. A simple weight decay can improve generalization. *Advances in Neural Information Processing Systems*, 4:950–957, 1992.

[231] L. Prechelt. Automatic early stopping using cross validation: Quantifying the criteria. *Neural Networks*, 11(4):761–767, 1998.

[232] G. Golub, M. Heath, G. Wahba. Generalised cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.

[233] D. J. C. Mackay. Probable networks and plausible predictions — a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6:469–505, 1995.

[234] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer Science and+Business Media LCC, New York, USA, 2006.

[235] D. J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.

[236] C. M. Bishop. Exact calculation of the hessian matrix for the multi-layer perceptron. *Neural Computation*, 4(4):494–501, 1992.

[237] C. M. Bishop. *Pattern Recognition and Machine Learning*, chapter 3.5.2 Maximizing the evidence function, pages 168–172. Springer Science and+Business Media LCC, New York, USA, 2006.

[238] D. J. C. MacKay. A practical Bayesian framework for backprop networks. In J. E. Moody, S. J. Hanson, R. P. Lippmann, editors, *Advances in Neural Information Processing Systems 4*, pages 839–846, 1992.

[239] V. Vapnik, editor. *Estimation of Dependences Based on Empirical Data [in Russian]*. Nauka, Moscow, 1979, English translation: 1982, Springer Verlag.

[240] C. J. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[241] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[242] J. C. Platt. *Advances in kernel methods: support vector learning*, chapter Fast training of support vector machines using sequential minimal optimization, pages 185–208. MIT Press, 1999.

[243] J. Sjöberg *et al.* Nonlinear black-box modeling in system identification: A unified overview. *Automatica*, 31(12):1691–1724, 1995.

[244] M. Friswell and J. Mottershead. *Finite Element Model Updating in Structural Dynamics*. Kluwer Academic Publishers, 1995.

[245] M. Bonne and A. Constantinescu. Inverse problems in elasticity. *Inverse Problems*, 21(2):R1–R50, 2005.

[246] S. Avril *et al.* Overview of identification methods of mechanical parameters based on full-field measurements. *Experimental Mechanics*, 48(4):381–402, 2008.

[247] P. S. Frederiksen. Experimental procedure and results for the identification of elastic constants of thick orthotropic plates. *Journal of Composite Materials*, 31(4):360–382, 1997.

[248] C. M. M. Soares, M. M. de Freitas, A. L. Araujo, P. Pedersen. Identification of material properties of composite plate specimens. *Composite Structures*, 25(1-4):277–285, 1993.

[249] A. A.L., C. Mota Soares, M. Moreira De Freitas. Characterization of material parameters of composite plate specimens using optimization and experimental vibrational data. *Composites Part B: Engineering*, 27(2 PART B):185–191, 1996.

[250] R. Mahnken and E. Stein. The identification of parameters for visco-plastic models via finite-element methods and gradient methods. *Modelling and Simulation in Materials Science and Engineering*, 2(3 A):597–616, 1994.

[251] J. C. Gelin and O. Ghouati. An inverse method for determining viscoplastic properties of aluminium alloys. *Journal of Materials Processing Tech.*, 45(1-4):435–440, 1994.

[252] N. Huber and C. Tsakmakis. A neural network tool for identifying the material parameters of a finite deformation viscoplasticity model with static recovery. *Computer Methods in Applied Mechanics and Engineering*, 191(3-5):353–384, 2001.

[253] M. Lefik and B. A. Schrefler. Artificial neural network for parameter identifications for an elasto-plastic model of superconducting cable under cyclic loading. *Computer Methods in Applied Mechanics and Engineering*, 192(28-30):3265–3283, 2003.

[254] C. C. Chang, T. Y. P. Chang, Y. G. Xu, M. L. Wang. Structural damage detection using an iterative neural network. *Journal of Intelligent Material Systems and Structures*, 11(1):32–42, 2000.

[255] C. Yun and E. Bahng. Substructural identification using neural networks. *Computers & Structures*, 77(1):41–52, 2000.

[256] S. F. Masri, A. W. Smyth, A. G. Chassiakos, T. K. Caughey, N. F. Hunter. Application of neural networks for detection of changes in nonlinear systems. *Journal of Engineering Mechanics*, 126(7):666–676, 2000.

[257] J. F. Unger, A. Teughels, G. De Roeck. Damage detection of a prestressed concrete beam using modal strains. *Journal of Structural Engineering*, 131(9):1456–1463, 2005.

[258] J. F. Unger, A. Teughels, G. De Roeck. System identification and damage detection of a prestressed concrete beam. *Journal of Structural Engineering*, 132(11):1691–1698, 2006.

[259] M. Brehm, V. Zabel, J. F. Unger. Stochastic model updating using perturbation methods combined with neural network estimations. In *IMAC-XXVII Conference & Exposition on Structural Dynamics*, Orlando,Florida, 2009.

[260] A. Kolmogorov. О представлении непрерывных функций нескольких переменных в виде суперпозиции непрерывных функций одного переменного, (On the representation of continuous functions of several variables by superpositions of continuous functions of one variable and addition). *Докл. АН СССР*, 114:679–681, 1957.

[261] H. Kupfer, H. Hilsdorf, H. Rusch. Behavior of concrete under biaxial stress. *ACI journal*, 66(8):656–666, 1966.

[262] S. Ghosh, K. Lee, P. Raghavan. A multi-level computational model for multi-scale damage analysis in composite and porous materials. *International Journal of Solids and Structures*, 38(14):2335–2385, 2001.

[263] S. Eckardt and C. Könke. Adaptive damage simulation of concrete using heterogeneous multiscale models. *Journal of Algorithms & Computational Technology*, 2(2):275–297, 2008.

[264] S. Ghosh, K. Lee, S. Moorthy. Two scale analysis of heterogeneous elastic-plastic materials with asymptotic homogenization and voronoi cell finite element model. *Computer Methods in Applied Mechanics and Engineering*, 132(1-2):63–116, 1996.

[265] I. Gitman. *Representative volumes and multi-scale modelling in quasi-brittle materials*. PhD thesis, Delft University of technology, 2006.

[266] I. M. Gitman, H. Askes, L. J. Sluys. Coupled-volume multi-scale modelling of quasi-brittle material. *European Journal of Mechanics, A/Solids*, 27(3):302–327, 2008.

[267] J. Ghaboussi, J. H. Garret, X. Wu. Material modeling with neural networks. In *Proceedings of the International conference on numerical methods in engineering: theory and applications, Swansea, UK*, pages 701–717, 1990.

[268] J. Ghaboussi, J. Garret, X. Wu. Knowledge-based modeling of material behavior with neural networks. *Journal of Engineering Mechanics Division, ASCE*, 117(1):132–153, 1991.

[269] Z. Waszczyszyn and L. Ziemianski. Neural networks in mechanics of structures and materials - new results and prospects of applications. *Computers & Structures*, 79(22-25):2261–2276, 2001.

[270] T. Furukawa and G. Yagawa. Implicit constitutive modelling for viscoplasticity using neural networks. *International Journal for Numerical Methods in Engineering*, 43(2):195–219, 1998.

[271] M. Lefik and B. A. Schrefler. Artificial neural network as an incremental non-linear constitutive model for a finite element code. *Computers & Structures*, 80(22):1699–1713, 2001.

[272] D. E. Sidarta and J. Ghaboussi. Constitutive modeling of geomaterials from non-uniform material tests. *Computers and Geotechnics*, 22(1):53–71, 1998.

[273] Q. Fu, Y. M. A. Hashash, S. Jung, J. Ghaboussi. Integration of laboratory testing and constitutive modeling of soils. *Computers and Geotechnics*, 34(5):330–345, 2007.

[274] J. Ghaboussi, D. Pecknold, M. Zhang, R. Haj-Ali. Autoprogressive training of neural network constitutive models. *Structural Engineering and Mechanics*, 42(1):105–126, 1998.

[275] R. Haj-Ali, D. A. Pecknold, J. Ghaboussi, G. Z. Voyiadjis. Simulated micromechanical models using artificial neural networks. *Journal of Engineering Mechancis*, 127(7):730–738, 2001.

[276] Y. Hashash, S. Jung, J. Ghaboussi. Numerical implementation of a neural network based material model in finite element analysis. *International Journal for Numerical Methods in Engineering*, 59:989–1005, 2004.

[277] J. Aboudi. *Mechanics of Composite Materials - A Unified Micromechanical Approach*. Elsevier, Amsterdam, 1991.

[278] J. F. Unger and C. Könke. Neural networks as material models within a multiscale approach. In B. H. Topping, editor, *Proceedings of the Ninth International Conference on the Application of Artificial Intelligence to Civil, Structural and Environmental Engineering*, St. Julians, Malta, 2007.

[279] P. Wolfe. A duality theorem for nonlinear programming. *Quarterly of Applied Mathematics*, 19(3):239–244, 1961.

Appendix A

Appendix to discrete crack models

A.1 Distance point ellipse

The shortest distance d between a point (u, v) and an ellipse can be reduced (by transformation) to a problem, where the ellipse is centered in the origin and its axes are aligned with the principal axes - the major radius a is aligned with the x-axis. Furthermore, the point is assumed to be in the first quadrant. Due to the symmetry of the problem, the shortest distance of a point in other quadrants can be reduced to the problem in the first quadrant, since $d(-u, v) = d(-u, -v) = d(-u, v) = d(u, v)$.

All points (x, y) on the ellipse fulfill

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 - 1 = 0, \tag{A.1}$$

where $a > b$. The shortest distance from (u, v) to a point (x, y) on the ellipse is described by a line that is normal to the ellipse, which can be expressed as

$$\nabla\left(\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 - 1\right) = \left(\frac{x}{a^2}, \frac{y}{b^2}\right) \tag{A.2}$$

$$(u - x, v - y) = t\left(\frac{x}{a^2}, \frac{y}{b^2}\right), \tag{A.3}$$

where t is a scaling parameter of the normal vector.

Case 1: $v = 0$

If the point lies on the origin $(u = 0, v = 0)$, the shortest distance to a point (x, y) on the ellipse is $d = -b$. Note that the distance is negative, which implies that the point (u, v) is inside the ellipse. If the point is moved to the right, it follows from the second coordinate in Eq. (A.3)

$$t = -b^2 \tag{A.4}$$

$$x = \frac{a^2 u}{a^2 - b^2} \tag{A.5}$$

Eq. (A.5) is obtained by substition of Eq. (A.4) into the first coordinate of Eq. (A.3) and then using Eq. (A.1). Since the point (x, y) lies on the ellipse, it is required that $x \leq a$ and, consequently,

$$\frac{a^2 u}{a^2 - b^2} \leq a, \tag{A.6}$$

which implies

$$u \leq \frac{a^2 - b^2}{a}. \tag{A.7}$$

As a result, the distance of a point $(u, 0)$ to the ellipse can be expressed as

$$d = \begin{cases} -\sqrt{(u-x)^2 + y^2} & u \leq \dfrac{a^2 - b^2}{a} \\ u - a & \text{otherwise} \end{cases} \tag{A.8}$$

where x and y are obtained from Eqs. (A.5) and (A.1):

$$y = b\sqrt{1 - \frac{x^2}{a^2}}. \tag{A.9}$$

Note that the distance is negative for a point inside the ellipse and positive otherwise.

Case 2: $v \neq 0$

Rearranging Eq. (A.3) and solving for x and y gives

$$x = \frac{ua^2}{t + a^2} \tag{A.10}$$

$$y = \frac{vb^2}{t + b^2}, \tag{A.11}$$

which is substituted into Eq. (A.1):

$$F(t) = 0 = \left(\frac{ua}{t + a^2}\right)^2 + \left(\frac{vb}{t + b^2}\right)^2 - 1. \tag{A.12}$$

The symmetry of the problem requires that $x > 0$ and $y > 0$ and from Eqs. (A.10) and (A.11) it follows that $t > -a^2$ and $t > -b^2$ and due to $a > b$ this finally gives

$$t > -b^2. \tag{A.13}$$

Solving Eq. (A.12) for t is analytically difficult, and an approximate solution using Newton's method is used. The first and second derivates of $F(t)$ are given as

$$\nabla F(t) = -\frac{2a^2u^2}{(t + a^2)^3} - \frac{2b^2v^2}{(t + b^2)^3} \tag{A.14}$$

$$\nabla^2 F(t) = \frac{6a^2u^2}{(t + a^2)^4} - \frac{6b^2v^2}{(t + b^2)^4}. \tag{A.15}$$

Due to Eq. (A.13), the denominators in Eqs. (A.14) and (A.15) are always positive and, consequently, $\nabla F(t) < 0$ and $\nabla^2 F(t) > 0$ $\forall t \epsilon (-b^2, \infty)$. This renders Newton's method an appropriate choice, since the function $F(t)$ is convex and strictly monotonic decreasing and

therefore possesses only a single solution. Starting with an initial value $t^{(0)} = b \cdot (v - b)$, the iterative series is obtained by

$$t^{(k+1)} = t^{(k)} - \frac{F(t^{(k)}}{\nabla F(t^{(k)})}. \tag{A.16}$$

Having solved Eq. (A.12) up to a certain accuracy, the point (x, y) on the ellipse can be calculated using Eqs. (A.10) and (A.11) and, finally, the distance d to (u, v) is obtained by:

$$d = \sqrt{(u - x)^2 + (v - y)^2} \tag{A.17}$$

$$= t\sqrt{\left(\frac{x}{a^2}\right)^2 + \left(\frac{y}{b^2}\right)^2} \tag{A.18}$$

Appendix B

Appendix to smeared crack models

B.1 Eigenvalues and derivatives (first and second) of symmetric tensor in 3D for the rounded Rankine criterion

B.1.1 Eigenvalues

The standard eigenvalue problem is given by

$$0 = \det \begin{pmatrix} s_0 - \sigma & s_3 & s_5 \\ s_3 & s_1 - \sigma & s_4 \\ s_5 & s_4 & s_2 - \sigma \end{pmatrix} \tag{B.1}$$

$$= \sigma^3 + a\sigma^2 + b\sigma + c \tag{B.2}$$

with

$$a = -\left(s_0 + s_1 + s_2\right) \tag{B.3}$$

$$b = s_0 s_1 + s_0 s_2 + s_1 s_2 - s_3^2 - s_4^2 - s_5^2 \tag{B.4}$$

$$c = s_0 s_4^2 + s_1 s_5^2 + s_2 s_3^2 - s_0 s_1 s_2 - 2 s_3 s_4 s_5. \tag{B.5}$$

By substituting $y = \sigma + a/3$, Eq. (B.2) can be rewritten as

$$0 = y^3 + 3py + 2q \tag{B.6}$$

$$q = \frac{a^3}{27} - \frac{ab}{6} + \frac{c}{2} \tag{B.7}$$

$$p = \frac{b}{3} - \frac{a^2}{9}. \tag{B.8}$$

The cubic equation has the following solution behavior as a function of the discriminate $D = p^3 + q^2$:

$D > 0$	one real and two conjugate complex roots
$D < 0$	three distinct real roots
$D = 0$, $q \neq 0$	two real roots, where one is of order 2
$D = 0$, $q = 0$	one real roots order 3.

For a symmetric matrix, the eigenvalues are always real and, as a result, $D \leq 0$ and, consequently, $p < 0$. In this case, the solution can be expressed as:

$$P = (\operatorname{sgn} q)\sqrt{|p|} \tag{B.9}$$

$$\beta = \frac{1}{3}\arccos\left(\frac{q}{P^3}\right) \tag{B.10}$$

$$\sigma_1 = -2P\cos(\beta) - \frac{a}{3} \tag{B.11}$$

$$\sigma_{2,3} = 2P\cos\left(\beta \pm \frac{\pi}{3}\right) - \frac{a}{3}. \tag{B.12}$$

B.1.2 One positive eigenvalue

For a single positive eigenvalue, which is either σ_1 or σ_3, the yield surface (Rankine) is given by

$$f_1 = \sigma_i \tag{B.13}$$

B.1.2.1 Gradient for distinct eigenvalues

For a single positive eigenvalue, the gradient can be calculated using the chain rule. In the following derivation the assumption of three distinct eigenvalues is made. The case of two identical eigenvalues is discussed in a second part.

$$\frac{\partial\sigma_1}{\partial s_i} = \frac{\partial\sigma_1}{\partial\beta}\frac{\partial\beta}{\partial s_i} + \frac{\partial\sigma_1}{\partial P}\frac{\partial P}{\partial s_i} + \frac{\partial\sigma_1}{\partial a}\frac{\partial a}{\partial s_i} \tag{B.14}$$

$$= 2P\sin(\beta)\frac{\partial\beta}{\partial s_i} - 2\cos(\beta)\frac{\partial P}{\partial s_i} - \frac{1}{3}\frac{\partial a}{\partial s_i} \tag{B.15}$$

$$\frac{\partial\sigma_{2,3}}{\partial s_i} = \frac{\partial\sigma_{2,3}}{\partial\beta}\frac{\partial\beta}{\partial s_i} + \frac{\partial\sigma_1}{\partial P}\frac{\partial P}{\partial s_i} + \frac{\partial\sigma_{2,3}}{\partial a}\frac{\partial a}{\partial s_i} \tag{B.16}$$

$$= -2P\sin(\beta \pm \frac{\pi}{3})\frac{\partial\beta}{\partial s_i} + 2\cos\left(\beta \pm \frac{\pi}{3}\right)\frac{\partial P}{\partial s_i} - \frac{1}{3}\frac{\partial a}{\partial s_i} \tag{B.17}$$

The derivatives of the auxiliary variables β, P, q and p with respect to the entries of the stress tensor are given by:

$$\frac{\partial\beta}{\partial s_i} = \frac{\partial\beta}{\partial q}\frac{\partial q}{\partial s_i} + \frac{\partial\beta}{\partial P}\frac{\partial P}{\partial s_i} \tag{B.18}$$

$$= -\frac{1}{3P^3\sqrt{1 - \frac{q^2}{P^6}}}\frac{\partial q}{\partial s_i} + \frac{q}{P^4\sqrt{1 - \frac{q^2}{P^6}}}\frac{\partial P}{\partial s_i} \tag{B.19}$$

$$\frac{\partial P}{\partial s_i} = \frac{\partial P}{\partial q}\frac{\partial q}{\partial s_i} + \frac{\partial P}{\partial p}\frac{\partial p}{\partial s_i} \tag{B.20}$$

$$= -\frac{\operatorname{sign} q}{2\sqrt{-p}}\frac{\partial p}{\partial s_i} \tag{B.21}$$

$$\frac{\partial q}{\partial s_i} = \frac{\partial q}{\partial a}\frac{\partial a}{\partial s_i} + \frac{\partial q}{\partial b}\frac{\partial b}{\partial s_i} + \frac{\partial q}{\partial c}\frac{\partial c}{\partial s_i} \tag{B.22}$$

$$= \left(\frac{a^2}{9} - \frac{b}{6}\right)\frac{\partial a}{\partial s_i} - \frac{a}{6}\frac{\partial b}{\partial s_i} + \frac{1}{2}\frac{\partial c}{\partial s_i} \tag{B.23}$$

$$\frac{\partial p}{\partial s_i} = \frac{\partial p}{\partial a}\frac{\partial a}{\partial s_i} + \frac{\partial p}{\partial b}\frac{\partial b}{\partial s_i} \tag{B.24}$$

$$= -\frac{2a}{9}\frac{\partial a}{\partial s_i} + \frac{1}{3}\frac{\partial b}{\partial s_i}. \tag{B.25}$$

Finally, the derivatives of the parameters a, b and c with respect to s_i are determined:

$$\frac{\partial a}{\partial s_i} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \frac{\partial b}{\partial s_i} = \begin{bmatrix} s_1 + s_2 \\ s_0 + s_2 \\ s_0 + s_1 \\ -2s_3 \\ -2s_4 \\ -2s_5 \end{bmatrix} \qquad \frac{\partial c}{\partial s_i} = \begin{bmatrix} s_4^2 - s_1 s_2 \\ s_5^2 - s_0 s_2 \\ s_3^2 - s_0 s_1 \\ 2(s_2 s_3 - s_4 s_5) \\ 2(s_0 s_4 - s_3 s_5) \\ 2(s_1 s_5 - s_3 s_4) \end{bmatrix}. \tag{B.26}$$

B.1.2.2 Hessian for distinct eigenvalues

The Hessian can be computed using the chain rule:

$$\begin{aligned} \frac{\partial^2 \sigma_1}{\partial s_i \partial s_j} = &\left(\frac{\partial^2 \sigma_1}{\partial^2 \beta}\frac{\partial \beta}{\partial s_j} + \frac{\partial^2 \sigma_1}{\partial \beta \partial P}\frac{\partial P}{\partial s_j} + \frac{\partial^2 \sigma_1}{\partial \beta \partial a}\frac{\partial a}{\partial s_j}\right)\frac{\partial \beta}{\partial s_i} + \frac{\partial \sigma_1}{\partial \beta}\frac{\partial^2 \beta}{\partial s_i \partial s_j} + \\ &\left(\frac{\partial^2 \sigma_1}{\partial^2 P}\frac{\partial P}{\partial s_j} + \frac{\partial^2 \sigma_1}{\partial \beta \partial P}\frac{\partial \beta}{\partial s_j} + \frac{\partial^2 \sigma_1}{\partial P \partial a}\frac{\partial a}{\partial s_j}\right)\frac{\partial P}{\partial s_i} + \frac{\partial \sigma_1}{\partial P}\frac{\partial^2 P}{\partial s_i \partial s_j} + \\ &\left(\frac{\partial^2 \sigma_1}{\partial^2 a}\frac{\partial a}{\partial s_j} + \frac{\partial^2 \sigma_1}{\partial \beta \partial a}\frac{\partial \beta}{\partial s_j} + \frac{\partial^2 \sigma_1}{\partial P \partial a}\frac{\partial P}{\partial s_j}\right)\frac{\partial a}{\partial s_i} + \frac{\partial \sigma_1}{\partial a}\frac{\partial^2 a}{\partial s_i \partial s_j} \end{aligned} \tag{B.27}$$

$$\begin{aligned} = &\left(2P\cos(\beta)\frac{\partial \beta}{\partial s_j} + 2\sin(\beta)\frac{\partial P}{\partial s_j}\right)\frac{\partial \beta}{\partial s_i} + 2P\sin(\beta)\frac{\partial^2 \beta}{\partial s_i \partial s_j} + \\ &\left(2\sin(\beta)\frac{\partial \beta}{\partial s_j}\right)\frac{\partial P}{\partial s_i} - 2\cos(\beta)\frac{\partial^2 P}{\partial s_i \partial s_j} + \frac{1}{3}\frac{\partial^2 a}{\partial s_i \partial s_j} \end{aligned} \tag{B.28}$$

$$\begin{aligned} \frac{\partial^2 \sigma_{2,3}}{\partial s_i \partial s_j} = &\left(\frac{\partial^2 \sigma_{2,3}}{\partial^2 \beta}\frac{\partial \beta}{\partial s_j} + \frac{\partial^2 \sigma_{2,3}}{\partial \beta \partial P}\frac{\partial P}{\partial s_j} + \frac{\partial^2 \sigma_{2,3}}{\partial \beta \partial a}\frac{\partial a}{\partial s_j}\right)\frac{\partial \beta}{\partial s_i} + \frac{\partial \sigma_{2,3}}{\partial \beta}\frac{\partial^2 \beta}{\partial s_i \partial s_j} + \\ &\left(\frac{\partial^2 \sigma_{2,3}}{\partial^2 P}\frac{\partial P}{\partial s_j} + \frac{\partial^2 \sigma_{2,3}}{\partial \beta \partial P}\frac{\partial \beta}{\partial s_j} + \frac{\partial^2 \sigma_{2,3}}{\partial a \partial P}\frac{\partial a}{\partial s_j}\right)\frac{\partial P}{\partial s_i} + \frac{\partial \sigma_{2,3}}{\partial P}\frac{\partial^2 P}{\partial s_i \partial s_j} + \\ &\left(\frac{\partial^2 \sigma_{2,3}}{\partial^2 a}\frac{\partial a}{\partial s_j} + \frac{\partial^2 \sigma_{2,3}}{\partial \beta \partial a}\frac{\partial \beta}{\partial s_j} + \frac{\partial^2 \sigma_{2,3}}{\partial a \partial P}\frac{\partial P}{\partial s_j}\right)\frac{\partial a}{\partial s_i} + \frac{\partial \sigma_{2,3}}{\partial a}\frac{\partial^2 a}{\partial s_i \partial s_j} \\ = &\left(-2P\cos\left(\beta \pm \frac{\pi}{3}\right)\frac{\partial \beta}{\partial s_j} - 2\sin\left(\beta \pm \frac{\pi}{3}\right)\frac{\partial P}{\partial s_j}\right)\frac{\partial \beta}{\partial s_i} - \\ &2P\sin\left(\beta \pm \frac{\pi}{3}\right)\frac{\partial^2 \beta}{\partial s_i \partial s_j} + \end{aligned}$$

$$\left(-2\sin\left(\beta\pm\frac{\pi}{3}\right)\frac{\partial\beta}{\partial s_j}\right)\frac{\partial P}{\partial s_i}+2\cos\left(\beta\pm\frac{\pi}{3}\right)\frac{\partial^2 P}{\partial s_i\partial s_j}+\frac{1}{3}\frac{\partial^2 a}{\partial s_i\partial s_j}.$$

The second derivatives of the auxiliary variables are given by:

$$\begin{aligned}\frac{\partial^2\beta}{\partial s_i\partial s_j}=&\left(\frac{\partial^2\beta}{\partial^2 q}\frac{\partial q}{\partial s_j}+\frac{\partial^2\beta}{\partial q\partial P}\frac{\partial P}{\partial s_j}\right)\frac{\partial q}{\partial s_i}+\frac{\partial\beta}{\partial q}\frac{\partial^2 q}{\partial s_i\partial s_j}+\\&\left(\frac{\partial^2\beta}{\partial^2 P}\frac{\partial P}{\partial s_j}+\frac{\partial^2\beta}{\partial q\partial P}\frac{\partial q}{\partial s_j}\right)\frac{\partial P}{\partial s_i}+\frac{\partial\beta}{\partial P}\frac{\partial^2 P}{\partial s_i\partial s_j}\end{aligned}\tag{B.29}$$

$$\begin{aligned}=&\left(-\frac{q}{3\left(P^3\sqrt{1-\frac{q^2}{P^6}}\right)^3}\frac{\partial q}{\partial s_j}+\frac{P^2}{\left(P^6-q^2\right)\sqrt{1-\frac{q^2}{P^6}}}\frac{\partial P}{\partial s_j}\right)\frac{\partial q}{\partial s_i}\\&-\frac{1}{3P^3\sqrt{1-\frac{q^2}{P^6}}}\frac{\partial^2 q}{\partial s_i\partial s_j}+\\&\left(-\frac{q\left(4P^6-q^2\right)}{\left(P^6-q^2\right)P^5\sqrt{1-\frac{q^2}{P^6}}}\frac{\partial P}{\partial s_j}+\frac{P^2}{\left(P^6-q^2\right)\sqrt{1-\frac{q^2}{P^6}}}\frac{\partial q}{\partial s_j}\right)\frac{\partial P}{\partial s_i}+\\&\frac{q}{P^4\sqrt{1-\frac{q^2}{P^6}}}\frac{\partial^2 P}{\partial s_i\partial s_j}.\end{aligned}\tag{B.30}$$

Remark: $P\neq 0$, since $D=p^3+q^2<0$(three different roots), the other cases are discussed separately. Furthermore $\left(1-\frac{q^2}{P^6}\right)\neq 0\leftrightarrow q^2\neq P^6\leftrightarrow D-p^3\neq|p^3|$, since $D<0$.

$$\begin{aligned}\frac{\partial^2 P}{\partial s_i\partial s_j}=&\left(\frac{\partial^2 P}{\partial^2 q}\frac{\partial q}{\partial s_j}+\frac{\partial^2 P}{\partial q\partial p}\frac{\partial p}{\partial s_j}\right)\frac{\partial q}{\partial s_i}+\frac{\partial P}{\partial q}\frac{\partial^2 q}{\partial s_i\partial s_j}+\\&\left(\frac{\partial^2 P}{\partial^2 p}\frac{\partial p}{\partial s_j}+\frac{\partial^2 P}{\partial q\partial p}\frac{\partial q}{\partial s_j}\right)\frac{\partial p}{\partial s_i}+\frac{\partial P}{\partial p}\frac{\partial^2 p}{\partial s_i\partial s_j}\end{aligned}\tag{B.31}$$

$$=-\left(\frac{\operatorname{sign} q}{4\sqrt{|p|}^3}\frac{\partial p}{\partial s_j}\right)\frac{\partial p}{\partial s_i}+\frac{\operatorname{sign} q\operatorname{sign} p}{2\sqrt{|p|}}\frac{\partial^2 p}{\partial s_i\partial s_j}\tag{B.32}$$

$$\begin{aligned}\frac{\partial^2 q}{\partial s_i\partial s_j}=&\left(\frac{\partial^2 q}{\partial^2 a}\frac{\partial a}{\partial s_j}+\frac{\partial^2 q}{\partial a\partial b}\frac{\partial b}{\partial s_j}+\frac{\partial^2 q}{\partial a\partial c}\frac{\partial c}{\partial s_j}\right)\frac{\partial a}{\partial s_i}+\frac{\partial q}{\partial a}\frac{\partial^2 a}{\partial s_i\partial s_j}+\\&\left(\frac{\partial^2 q}{\partial a\partial b}\frac{\partial a}{\partial s_j}+\frac{\partial^2 q}{\partial^2 b}\frac{\partial b}{\partial s_j}+\frac{\partial^2 q}{\partial b\partial c}\frac{\partial c}{\partial s_j}\right)\frac{\partial b}{\partial s_i}+\frac{\partial q}{\partial b}\frac{\partial^2 b}{\partial s_i\partial s_j}+\\&\left(\frac{\partial^2 q}{\partial a\partial c}\frac{\partial a}{\partial s_j}+\frac{\partial^2 q}{\partial b\partial c}\frac{\partial b}{\partial s_j}+\frac{\partial^2 q}{\partial^2 c}\frac{\partial c}{\partial s_j}\right)\frac{\partial c}{\partial s_i}+\frac{\partial q}{\partial c}\frac{\partial^2 c}{\partial s_i\partial s_j}\end{aligned}\tag{B.33}$$

$$= \left(\frac{2a}{9}\frac{\partial a}{\partial s_j} - \frac{1}{6}\frac{\partial b}{\partial s_j}\right)\frac{\partial a}{\partial s_i} + \left(\frac{a^2}{9} - \frac{b}{6}\right)\frac{\partial^2 a}{\partial s_i \partial s_j}$$
$$\left(-\frac{1}{6}\frac{\partial a}{\partial s_j}\right)\frac{\partial b}{\partial s_i} - \frac{a}{6}\frac{\partial^2 b}{\partial s_i \partial s_j} + \frac{1}{2}\frac{\partial^2 c}{\partial s_i \partial s_j}$$

$$\frac{\partial^2 p}{\partial s_i \partial s_j} = \left(\frac{\partial^2 p}{\partial^2 a}\frac{\partial a}{\partial s_j} + \frac{\partial^2 p}{\partial a \partial b}\frac{\partial b}{\partial s_j}\right)\frac{\partial a}{\partial s_i} + \frac{\partial p}{\partial a}\frac{\partial^2 a}{\partial s_i \partial s_j} +$$
$$\left(\frac{\partial^2 p}{\partial a \partial b}\frac{\partial a}{\partial s_j} + \frac{\partial^2 p}{\partial^2 b}\frac{\partial b}{\partial s_j}\right)\frac{\partial b}{\partial s_i} + \frac{\partial p}{\partial b}\frac{\partial^2 b}{\partial s_i \partial s_j} \tag{B.34}$$
$$= -\frac{2}{9}\frac{\partial a}{\partial s_i}\frac{\partial a}{\partial s_j} - \frac{2a}{9}\frac{\partial^2 a}{\partial s_i \partial s_j} + \frac{1}{3}\frac{\partial^2 b}{\partial s_i \partial s_j} \tag{B.35}$$

$$\frac{\partial^2 a}{\partial s_i \partial s_j} = 0 \tag{B.36}$$

$$\frac{\partial^2 b}{\partial s_i \partial s_j} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 \end{bmatrix} \tag{B.37}$$

$$\frac{\partial^2 c}{\partial s_i \partial s_j} = \begin{bmatrix} 0 & -s_2 & -s_1 & 0 & 2s_4 & 0 \\ -s_2 & 0 & -s_0 & 0 & 0 & 2s_5 \\ -s_1 & -s_0 & 0 & 2s_3 & 0 & 0 \\ 0 & 0 & 2s_3 & 2s_2 & -2s_5 & -2s_4 \\ 2s_4 & 0 & 0 & -2s_5 & 2s_0 & -2s_3 \\ 0 & 2s_5 & 0 & -2s_4 & -2s_3 & 2s_1 \end{bmatrix}. \tag{B.38}$$

B.1.2.3 Gradient and Hessian for two identical eigenvalues

Identical eigenvalues are obtained in the case of $D = 0$, which is equivalent to $\beta = 0$. As a result, the identical eigenvalues are σ_2 and σ_3. Since only one eigenvalue is positive, it has to be σ_1, which is a function of P, q, a. In this special case the gradient and the Hessian can be expressed as

$$\lim_{q \to P^3} \frac{\partial \sigma_1}{\partial q} = -\frac{2}{9P^2} \tag{B.39}$$
$$\lim_{q \to P^3} \frac{\partial \sigma_1}{\partial P} = -\frac{4}{3} \tag{B.40}$$
$$\lim_{q \to P^3} \frac{\partial \sigma_1}{\partial a} = -\frac{1}{3} \tag{B.41}$$

and the Hessian is given by

$$\lim_{q\to P^3} \frac{\partial^2 \sigma_1}{\partial^2 q} = \frac{16}{243P^5} \tag{B.42}$$

$$\lim_{q\to P^3} \frac{\partial^2 \sigma_1}{\partial q \partial P} = \frac{20}{81P^3} \tag{B.43}$$

$$\lim_{q\to P^3} \frac{\partial^2 \sigma_1}{\partial^2 P} = -\frac{20}{27P}. \tag{B.44}$$

The second derivatives with respect to a vanish. As a result, the gradient and Hessian can be computed as

$$\frac{\partial \sigma_1}{\partial s_i} = \frac{\partial \sigma_1}{\partial q}\frac{\partial q}{\partial s_i} + \frac{\partial \sigma_1}{\partial P}\frac{\partial P}{\partial s_i} + \frac{\partial \sigma_1}{\partial a}\frac{\partial a}{\partial s_i} \tag{B.45}$$

$$\begin{aligned}\frac{\partial^2 \sigma_1}{\partial s_i \partial s_j} =& \left(\frac{\partial^2 \sigma_1}{\partial^2 q}\frac{\partial q}{\partial s_j} + \frac{\partial^2 \sigma_1}{\partial q \partial P}\frac{\partial P}{\partial s_j}\right)\frac{\partial q}{\partial s_i} + \frac{\partial \sigma_1}{\partial q}\frac{\partial^2 q}{\partial s_i \partial s_j} + \\ & \left(\frac{\partial^2 \sigma_1}{\partial^2 P}\frac{\partial P}{\partial s_j} + \frac{\partial^2 \sigma_1}{\partial q \partial P}\frac{\partial q}{\partial s_j}\right)\frac{\partial P}{\partial s_i} + \frac{\partial \sigma_1}{\partial P}\frac{\partial^2 P}{\partial s_i \partial s_j},\end{aligned} \tag{B.46}$$

where the fact that the Hessian with respect to a is zero has been used.

B.1.3 Two positive eigenvalues

For two positve eigenvalues σ_k, σ_l the yield surface is of the rounded Rankine type

$$f_2 = \sqrt{\sigma_k^2 + \sigma_l^2} \tag{B.47}$$

B.1.3.1 Gradient for distinct eigenvalues

In the case of distinct eigenvalues, the gradient can be determined using the chain rule:

$$\frac{\partial f_2}{\partial s_i} = \frac{1}{f_2}\left(\sigma_k \frac{\partial \sigma_k}{\partial s_i} + \sigma_l \frac{\partial \sigma_l}{\partial s_i}\right) \tag{B.48}$$

B.1.3.2 Hessian for distinct eigenvalues

By further application of the chain rule, the Hessian can be expressed as:

$$\begin{aligned}\frac{\partial^2 f_2}{\partial s_i \partial s_j} =& \frac{1}{f_2^3}\left[\frac{\partial^2 \sigma_k}{\partial s_i \partial s_j}\sigma_k f_2^2 + \frac{\partial^2 \sigma_l}{\partial s_i \partial s_j}\sigma_l f_2^2 + \right. \\ & \left. \frac{\partial \sigma_k}{\partial s_i}\left(\sigma_l^2 \frac{\partial \sigma k}{\partial s_j} - \sigma_k \sigma_l \frac{\partial \sigma l}{\partial s_j}\right) + \frac{\partial \sigma_l}{\partial s_i}\left(\sigma_k^2 \frac{\partial \sigma l}{\partial s_j} - \sigma_k \sigma_l \frac{\partial \sigma k}{\partial s_j}\right)\right].\end{aligned} \tag{B.49}$$

B.1.3.3 Gradient and Hessian for two identical real roots

One real root is of order 2. This is equal to $D = 0, q \neq 0$. It follows directly that

$$\frac{q}{P^3} = 1 \leftrightarrow \beta = 0. \tag{B.50}$$

As a result, only the yield functions σ_2 and σ_3 are relevant and the root of order 2 is given by

$$\sigma_{2,3} = 2P\cos\left(\frac{\pi}{3}\right) - \frac{a}{3}. \tag{B.51}$$

For positive $\sigma_{2,3}$, the gradient and the Hessian of β with respect to q and P cannot be calculated with Eqs. (B.18) and (B.29) due to the vanishing divisor. For this special case the derivatives are calculated as follows. The second and third principal stress are combined in a modified (rounded) Rankine criterion:

$$f_2 = \sqrt{\sigma_2^2 + \sigma_3^2} \tag{B.52}$$

$$= \sqrt{2P^2\cos^2\beta + 6P^2\sin^2\beta + \frac{2a^2}{9} - \frac{4a}{3}P\cos\beta} \tag{B.53}$$

In an analogue way the derivatives are calculated

$$\lim_{q\to P^3}\frac{\partial f_2}{\partial q} = -\frac{\sqrt{2}}{9}\frac{(6P+a)}{|3P-a|P^2} \tag{B.54}$$

$$\lim_{q\to P^3}\frac{\partial f_2}{\partial P} = \frac{\sqrt{2}}{3}\frac{(15P-2a)}{|3P-a|} \tag{B.55}$$

$$\lim_{q\to P^3}\frac{\partial f_2}{\partial a} = \operatorname{sign}(3P-a)\frac{\sqrt{2}}{3} \tag{B.56}$$

and the Hessian is given by

$$\lim_{q\to P^3}\frac{\partial^2 f_2}{\partial^2 q} = -\frac{\sqrt{2}}{243}\frac{(54P^3 - 216P^2a + 27Pa^2 + 8a^3)}{P^5\left|3P-a\right|^3} \tag{B.57}$$

$$\lim_{q\to P^3}\frac{\partial^2 f_2}{\partial q\partial P} = \frac{\sqrt{2}}{81}\frac{(1026P^3 + 27P^2a - 54Pa^2 - 10a^3)}{P^3\left|3P-a\right|^3} \tag{B.58}$$

$$\lim_{q\to P^3}\frac{\partial^2 f_2}{\partial^2 P} = -\frac{\sqrt{2}}{27}\frac{(1026P^3 - 216P^2a - 135Pa^2 - 10a^3)}{P\left|3P-a\right|^3}. \tag{B.59}$$

Using the partial derivatives, the computation of the gradient and Hessian is analogue to Eqs. (B.45) and (B.46). Remark: $P \neq 0$, since $P = \operatorname{sign}(q)\sqrt{(}-p)$ and $p < 0$. Furthermore $3P - a \neq 0$. The proof is as follows. Assume $a = 3P$. As a result

$$p = \frac{b}{3} - \frac{a^2}{9} \tag{B.60}$$

$$= \frac{b}{3} - \frac{9P^2}{9} \tag{B.61}$$

$$= \frac{b}{3} + p. \tag{B.62}$$

As a result, it follows that $b = 0$. Furthermore holds:

$$D = 0 \tag{B.63}$$
$$= p^3 + q^2 \tag{B.64}$$
$$= -\frac{a^6}{729} + \frac{a^6}{729} + \frac{a^3}{27}c + \frac{c^2}{4} \tag{B.65}$$
$$= \frac{c}{108}\left(4a^3 + 27c\right). \tag{B.66}$$

If $c = 0$, the characteristic equation Eq. (B.2) reduces to

$$\sigma^3 + a\sigma^2 = 0, \tag{B.67}$$

which has the solutions $0, 0, -a$. This is in contradiction to the assumption of two positve eigenvalues. As a result

$$c = -\frac{4a^3}{27} \tag{B.68}$$
$$q = -\frac{a^3}{27} \tag{B.69}$$
$$P = \text{sign}\left(-\frac{a^3}{27}\right)\frac{|a|}{3} \tag{B.70}$$
$$= \text{sign}\left(-\frac{a^4}{27}\right)\frac{a}{3} \tag{B.71}$$
$$= -\frac{a}{3}. \tag{B.72}$$

This is in contradiction to the assumption $3P = a$ at the beginning of the proof. As a result, $3P - a \neq 0$ and, consequently, the terms in Eqs. (B.54)-(B.59) can be computed.

B.1.4 Three positive eigenvalues

For three positve eigenvalues $\sigma_1, \sigma_2, \sigma_3$, the yield surface is of the rounded Rankine type

$$f_3 = \sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2} \tag{B.73}$$

Using Eqs. (B.11) and (B.12), the yieldsurface can be further simplified:

$$\sigma_1^2 = 4P^2\cos^2(\beta) + \frac{a^2}{9} + \frac{4a}{3}P\cos\beta \tag{B.74}$$
$$\sigma_2^2 = P^2\cos^2(\beta) + 3P^2\sin^2\beta + \frac{a^2}{9} + 2\left[-P^2\sqrt{3}\cos\beta\sin\beta - \frac{a}{3}P\cos\beta + \frac{\sqrt{3}a}{3}P\sin\beta\right] \tag{B.75}$$
$$\sigma_3^2 = P^2\cos^2(\beta) + 3P^2\sin^2\beta + \frac{a^2}{9} + 2\left[P^2\sqrt{3}\cos\beta\sin\beta - \frac{a}{3}P\cos\beta - \frac{\sqrt{3}a}{3}P\sin\beta\right], \tag{B.76}$$

where the theorem $\cos(\alpha \pm \beta) = \cos\alpha\cos\beta \mp \sin\alpha\sin\beta$ has been used. As a result the yield function f_3 is given by:

$$f_3 = \sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2} \tag{B.77}$$

$$= \sqrt{6P^2\left(\cos^2\beta + \sin^2\beta\right) + \frac{a^2}{3}} \tag{B.78}$$

$$= \sqrt{6P^2 + \frac{a^2}{3}} \tag{B.79}$$

$$= \sqrt{-6p + \frac{a^2}{3}} \tag{B.80}$$

$$= \sqrt{a^2 - 2b}, \tag{B.81}$$

where the condition that $p \leq 0$ has been used (since the discriminante $D = p^3 + q^2 \leq 0$, because a symmetric matrix has only real eigenvalues). As a result, the gradient and the Hessian can be expressed as

$$\frac{\partial f_3}{\partial s_i} = \frac{\partial f_3}{\partial a}\frac{\partial a}{\partial s_i} + \frac{\partial f_3}{\partial b}\frac{\partial b}{\partial s_i} \tag{B.82}$$

$$\begin{aligned}\frac{\partial^2 f_3}{\partial s_i \partial s_j} = &\left(\frac{\partial^2 f_3}{\partial^2 a}\frac{\partial a}{\partial s_j} + \frac{\partial^2}{\partial a \partial b}\frac{\partial b}{\partial s_j}\right)\frac{\partial a}{\partial s_i} + \frac{\partial f_3}{\partial a}\frac{\partial^2 a}{\partial s_i \partial s_j} + \\ &\left(\frac{\partial^2 f_3}{\partial a \partial b}\frac{\partial a}{\partial s_j} + \frac{\partial^2 f_3}{\partial^2 b}\frac{\partial b}{\partial s_j}\right)\frac{\partial b}{\partial s_i} + \frac{\partial f_3}{\partial b}\frac{\partial^2 b}{\partial s_i \partial s_j}\end{aligned} \tag{B.83}$$

where

$$\frac{\partial f_3}{\partial a} = \frac{a}{f_3} \tag{B.84}$$

$$\frac{\partial f_3}{\partial b} = -\frac{1}{f_3} \tag{B.85}$$

$$\frac{\partial^2 f_3}{\partial^2 a} = -\frac{2b}{f_3^3} \tag{B.86}$$

$$\frac{\partial^2 f_3}{\partial a \partial b} = -\frac{a}{f_3^3} \tag{B.87}$$

$$\frac{\partial^2 f_3}{\partial^2 b} = \frac{1}{f_3^2} \tag{B.88}$$

Appendix C

Appendix to meta models

C.1 Standard form of the Gaussian distribution

The probability density function of a Gaussian distribution is given by

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{D}{2}}}\frac{1}{|\boldsymbol{\Sigma}|^{\frac{1}{2}}}e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}, \tag{C.1}$$

where D is the dimension of $\boldsymbol{x}$ and $|\boldsymbol{\Sigma}|$ is the determinant of $\boldsymbol{\Sigma}$.

C.2 Completing the square

For a Gaussian distribution, the exponent terms can be expressed as

$$-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu}\right)^T\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu}\right) = -\frac{1}{2}\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \text{const.} \tag{C.2}$$

Given a quadratic form of the exponent terms, the covariance matrix $\boldsymbol{\Sigma}$ can be determined by comparing the pure quadratic terms, and afterwards, the mean $\boldsymbol{\mu}$ is expressed as a function of the covariance matrix and the linear terms.

C.3 Linear Algebra

C.3.1 Matrix inversion

$$\left(\boldsymbol{A}+\boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}\right)^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\left(\boldsymbol{D}+\boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B}\right)^{-1}\boldsymbol{C}\boldsymbol{A}^{-1} \tag{C.3}$$

This is known as the Woodbury identity, which can be verified by multiplying both sides with $\boldsymbol{A}+\boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{C}$.

C.3.2 Determinant of product of two matrices

The determinant of a product of two matrices is identical to the product of the determinants of each matrix.

$$|\boldsymbol{A}\boldsymbol{B}| = |\boldsymbol{A}|\,|\boldsymbol{B}| \tag{C.4}$$

C.3.3 Determinant of sum of two matrices

Given two matrices $\boldsymbol{A}, \boldsymbol{B}$ of size $N \times M$, it follows that

$$\left|\boldsymbol{I}_N + \boldsymbol{A}\boldsymbol{B}^T\right| = \left|\boldsymbol{I}_M + \boldsymbol{A}^T\boldsymbol{B}\right|. \tag{C.5}$$

If $\boldsymbol{A}$ and $\boldsymbol{B}$ are vectors, this further simplifies to

$$\left|\boldsymbol{I}_N + \boldsymbol{a}\boldsymbol{b}^T\right| = 1 + \boldsymbol{a}^T\boldsymbol{b}. \tag{C.6}$$

C.3.4 Derivative of logarithm of determinant of a matrix

This formula is derived e.g. in [234] in the appendix:

$$\frac{\partial}{\partial x} \ln |\boldsymbol{A}| = \operatorname{tr}\left(\boldsymbol{A}^{-1}\frac{\partial \boldsymbol{A}}{\partial x}\right). \tag{C.7}$$

C.4 Support vector machines

C.4.1 Wolfe's dual for a quadratic convex program

The primal problem is given by

$$\max\ f(\boldsymbol{x}) \tag{C.8}$$

$$\text{subject to } g_i(\boldsymbol{x}) \leq 0 \quad \forall i = 1..N, \tag{C.9}$$

where N corresponds to the number of inequality constraints. The corresponding dual problem is given by:

$$\min\ f(\boldsymbol{x}) - \sum_{i=1}^{N} \lambda_i g_i(\boldsymbol{x}) \tag{C.10}$$

$$\text{subject to } \lambda_i \geq 0 \quad \forall i \tag{C.11}$$

$$\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} = \sum_{i=1}^{N} \lambda_i \frac{\partial g_i(\boldsymbol{x})}{\partial \boldsymbol{x}}. \tag{C.12}$$

If the objective function of the primal is concave and the constraints $g_i(\boldsymbol{x})$ are convex, the optimal solution of the primal problem is identical to the optimal solution of the dual problem.

C.4.2 Application of Wolfe's dual to the linear separable case of support vector machines

The primal problem is given by:

$$\max\ -\frac{1}{2}\|\boldsymbol{w}\|^2 \tag{C.13}$$

$$\text{subject to } 1 - y_i\left(\boldsymbol{x}_i\boldsymbol{w} + b\right) \leq 0 \quad \forall i = 1..L. \tag{C.14}$$

Using Eq. (C.10), the corresponding dual problem can be expressed as:

$$\min \; -\frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{L}\alpha_i \left[1 - y_i \left(\boldsymbol{x}_i\boldsymbol{w} + b\right)\right] \tag{C.15}$$

$$\text{subject to } \alpha_i \geq 0 \quad \forall i \tag{C.16}$$

$$\frac{\partial f(\boldsymbol{w}, b)}{\partial \boldsymbol{w}} = \sum_{i=1}^{L}\alpha_i \frac{\partial g_i(\boldsymbol{w}, b)}{\partial \boldsymbol{w}} \tag{C.17}$$

$$\frac{\partial f(\boldsymbol{w}, b)}{\partial b} = \sum_{i=1}^{L}\alpha_i \frac{\partial g_i(\boldsymbol{w}, b)}{\partial b}. \tag{C.18}$$

From Eqs. (C.17) and (C.18) it follows:

$$\boldsymbol{w} = \sum_{i=1}^{L}\alpha_i y_i \boldsymbol{x}_i \tag{C.19}$$

$$0 = \sum_{i=1}^{L}\alpha_i y_i. \tag{C.20}$$

Substituting Eq. (C.19) into Eq. (C.15) gives:

$$\min \; -\frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i\boldsymbol{x}_j - \sum_{i=1}^{L}\alpha_i - \sum_{i=1}^{L}\alpha_i y_i b + \sum_{i=1}^{L}\sum_{j=1}^{L}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i\boldsymbol{x}_j. \tag{C.21}$$

Further substition of Eq. (C.20) into Eq. (C.21) and changing the sign finally gives

$$\max \; -\frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i\boldsymbol{x}_j + \sum_{i=1}^{L}\alpha_i \tag{C.22}$$

$$\text{subject to } \alpha_i \geq 0 \quad \forall i = 1..L \tag{C.23}$$

$$0 = \sum_{i=1}^{L}\alpha_i y_i \tag{C.24}$$

$$\boldsymbol{w} = \sum_{i=1}^{L}\alpha_i y_i \boldsymbol{x}_i. \tag{C.25}$$

C.4.3 Application of Wolfe's dual to the linear non separable case of support vector machines

The primal problem is given by:

$$\max \; -\frac{1}{2}\|\boldsymbol{w}\|^2 - C\sum_{i=1}^{L}\xi_i \tag{C.26}$$

$$\text{subject to } g_i^1 : \; 1 - \xi_i - y_i \left(\boldsymbol{x}_i\boldsymbol{w} + b\right) \leq 0 \quad \forall i = 1..L \tag{C.27}$$

$$g_i^2 : \; -\xi_i \leq 0 \quad \forall i = 1..L. \tag{C.28}$$

Using Eq. (C.10), the corresponding dual problem can be expressed as [279]:

$$\min \; -\frac{1}{2}\|\boldsymbol{w}\|^2 - C\sum_{i=1}^{L}\xi_i - \sum_{i=1}^{L}\alpha_i\left[1-\xi_i - y_i\left(\boldsymbol{x}_i\boldsymbol{w}+b\right)\right] + \sum_{i=1}^{L}\beta_j\xi_j \tag{C.29}$$

$$\text{subject to } \alpha_i \geq 0 \quad \forall i \tag{C.30}$$

$$\beta_i \geq 0 \quad \forall i \tag{C.31}$$

$$\frac{\partial f(\boldsymbol{w},b,\boldsymbol{\xi})}{\partial \boldsymbol{w}} = \sum_{i=1}^{L}\alpha_i\frac{\partial g_i^1(\boldsymbol{w},b)}{\partial \boldsymbol{w}} \tag{C.32}$$

$$\frac{\partial f(\boldsymbol{w},b,\boldsymbol{\xi})}{\partial b} = \sum_{i=1}^{L}\alpha_i\frac{\partial g_i^1(\boldsymbol{w},b)}{\partial b} \tag{C.33}$$

$$\frac{\partial f(\boldsymbol{w},b,\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} = \sum_{i=1}^{L}\alpha_i\frac{\partial g_i^1(\boldsymbol{w},b,\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} + \sum_{i=1}^{L}\beta_i\frac{\partial g_i^2(\boldsymbol{w},b,\boldsymbol{\xi})}{\partial \boldsymbol{\xi}}, \tag{C.34}$$

where α_i are the Lagrange multipliers corresponding to constraints in Eq. (C.27), whereas β_i are the Lagrange multipliers corresponding to the non negativity constraints in Eq. (C.28). From Eqs. (C.32), (C.33), (C.34) it follows:

$$\boldsymbol{w} = \sum_{i=1}^{L}\alpha_i y_i \boldsymbol{x}_i \tag{C.35}$$

$$0 = \sum_{i=1}^{L}\alpha_i y_i \tag{C.36}$$

$$-C = -\alpha_i - \beta_i. \tag{C.37}$$

Substitution of Eqs. (C.35) and (C.37), resolved with respect to β_i, into Eq. (C.29), the objective function can be written as

$$\begin{aligned}\min \; -\frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i\boldsymbol{x}_j - C\sum_{i=1}^{L}\xi_i - \sum_{i=1}^{L}\alpha_i + \sum_{i=1}^{L}\alpha_i\xi_i \\ + \sum_{i=1}^{L}\alpha_i y_i b + \sum_{i=1}^{L}\sum_{j=1}^{L}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i\boldsymbol{x}_j + \sum_{i=1}^{L}(C-\alpha_i)\xi_i.\end{aligned} \tag{C.38}$$

Further substition of Eq. (C.36) into Eq. (C.38) and changing the sign finally gives the optimization problem :

$$\max \; -\frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i\boldsymbol{x}_j + \sum_{i=1}^{L}\alpha_i \tag{C.39}$$

$$\text{subject to } \; 0 \leq \alpha_i \leq C \quad \forall i = 1..L \tag{C.40}$$

$$\boldsymbol{w} = \sum_{i=1}^{L}\alpha_i y_i \boldsymbol{x}_i \tag{C.41}$$

$$0 = \sum_{i=1}^{L} \alpha_i y_i \tag{C.42}$$

$$\beta_i = C - \alpha_i. \tag{C.43}$$

The condition $C \leq \alpha_i$ is derived from Eqs. (C.37) and (C.31) and replaces the condition $\beta_i \geq 0$. It is to be noted that the slack variables and its Lagrange multipliers disappear in the the dual problem.