Bergische Universität Wuppertal

Shape Sensitivities for the Failure Probability of Mechanical Components

Dissertation

submitted to the Faculty of Mathematics and Science

May, 2021

By: Mohamed Saadi
First Supervisor : Prof. Dr. Hanno Gottschalk
Second Supervisor: Prof. Dr. Matthias Bolten

The PhD thesis can be quoted as follows:

urn:nbn:de:hbz:468-20210526-110304-6
[http://nbn-resolving.de/urn/resolver.pl?urn=urn%3Anbn%3Ade%3Ahbz%3A468-20210526-110304-6]

DOI: 10.25926/dp9r-hj27
[https://doi.org/10.25926/dp9r-hj27]

Acknowledgments

The major part of this thesis was written while I was an employee at the Faculty of Mathematics and Computer Science at the University of Wuppertal.

First of all, I would like to thank Almighty Allah for granting me this opportunity, determination, strength and patience to complete my dissertation, despite all challenges and difficulties, to the end.

I would like to express my sincere and greatest gratitude to my advisor Prof. Hanno Gotschalk for the continuous support during all the way of my Ph.D study and related research, for his patience, motivation, and his immense knowledge. His guidance helped me during the time of research till writing this thesis. I could not have imagined getting a better advisor and mentor for my Ph.D study.

Thanks are also to Prof. Dr. Matthias Bolten, Prof. Dr. K. Klamroth and Prof. Matthias Ehrhardt for their acceptance being as members of the examination committee.

I gratefully acknowledge the funding sources that made my Ph.D work possible. This research-work has been funded by Siemens AG Energy and the German federal ministry of economic affairs BMWi via an AG Turbo grant.

Last but not least, I would like to thank my family: my parents, my brothers and sisters for supporting me spiritually throughout writing this thesis and my life in general.

Finally, I would like to take this opportunity to thank all my friends and colleagues who have given their support and help as well.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

During recent years, the methods used in sensitivity analysis of complex engineering problems, such as turbomachinery design, have developed rapidly with the demand for more reliable components. Furthermore, the public interest is focusing on concepts that improve the energy efficiency of power and heat supply systems with low energy cost. This increases the necessity for innovative design approaches for the construction of robust mechanical components for gas turbine power plant. While the traditional methods were based on simplified theories, experience and intuition of engineers, new design methods use computational fluid dynamics and computational structural mechanics based on numerical simulation and optimization algorithms.

From an industrial point of view, the necessity exists to improve structural integrity of engineering components with reduction of design time and cost of products. Therefore, the design process can not be satisfactorily carried out without considering advanced life prediction methods. The formation and growth of cracks under cyclic loading (fatigue) in metal is a random process [16]. The properties and conditions of materials in any structure subjected to cyclic loads will vary in a random manner [16]. In the gas turbine industry for example, increasing the mass flows and the temperature loads of gas-fired power plants leads to higher and higher mechanical and thermal load on the blading. Due to higher turbine inlet temperatures, the centrifugal forces increase with larger turbine radii as well as thermal loads. Existing material reserves will thus be increasingly exploited and a detailed probabilistic assessment of the risk of failure will be crucial to guarantee the safe operation of combined cycle power plants. Furthermore, the volatile supply of wind and solar energy into the electricity grid poses new challenges to gas-fired power plants with regard to starting frequency and thus increases the relevance of low-cycle fatigue (LCF), which is subjected to an inherent scattering.

For economic reasons, there is a need to integrate the above-mentioned scatter into the methods of design. In addition to the material scatter, manufacturing tolerances also become drivers of the expected service life, which often leads to deviations that are quite relevant in terms of the design life.

Modern 3D design methods, especially in the context of automated optimization of aerodynamic design, are the drivers of further efficiency gains (beyond 60% efficiency rate) in combined cycle power plants. Thereby, losses can in many cases be significantly reduced. Examples of efficiency gains up to a few tenths of a percent by individual components can be found in the literatures [20, 32, 37, 53]. In practice, it becomes often apparent that monodisciplinary optimization - purely aerodynamic optimization - usually do not lead to realizable designs, or that these designs have to be modified in such a way that a significant part of the efficiency increase is lost again. A necessary consequence of this observation is an application of multi-criteria optimization strategies, in which the mechanical and aerodynamic performance of components in synchronous simulations is assessed [33, 49].

1.2 The Objectives of This Ph.D. Project

The aim of this Ph.D. project is the development of methods and design tools to determine the sensitivity of the probability of failure of mechanically and thermally stressed hot gas components under the variation of design parameters and production-related shape deviations. Based on the research findings of a predecessor project 'Probabilistische Lebensdauerberechnung für Design bei extremen Temperaturen (AG Turbo Project 4.1.2. [58]) which describes the basic methodology for calculating default risks, the current project focuses on design techniques resulting from this approach.

A local and probabilistic model for LCF in the context of polycrystalline metal based on an appropriate combination of reliability statistics and Poisson point process has been developed in [43]. This probabilistic model quantitatively describes the failure mechanism of surface driven LCF in terms of the following Weibull cumulative distribution function:

$$F_N(t) = 1 - \exp\left[- \left(\frac{t}{\eta}\right)^{\bar{m}} \right] \tag{1.1}$$

where

$$\eta = \left(\int_{\partial\Omega} \frac{1}{N_{det}^{\bar{m}}} dA \right)^{-1/\bar{m}} . \tag{1.2}$$

Here stand $\partial\Omega$ for the boundary of a bounded region $\Omega \subseteq \mathbb{R}^3$ filled with some polycrystalline metal, t for the number of load cycles, N_{det} for number of cycles to crack initiation, $\bar{m}$ and η for Weibull shape and scale parameters respectively.

In contrast to the deterministic design philosophy, probabilistic design not only assesses the highest point of loading in a component, but also the potential crack initiation in less stressed locations. As a side effect, it is to be noted that the objective functional, i.e. which is the probability of survival for the duration of a given number of cycles or number of operating hours, take the form of integrals over local stress and temperature fields. This local integral form of the probabilistic objective functional assimilates both structure-mechanical and fluid dynamical design and leads to functional differentiability of the objective functionals according to the temperature and stress fields.

The smooth variation of the probabilistic objective functional within the design space encourages the use of gradient-based shape optimization approach. The advantage of such method is the convergence to a load optimum with a significantly smaller number of functional evaluations [29]. Since the number of design variables, which represent the surface perturbation of a mechanical component, is too large, an effective computation of the shape gradients necessities the integration of the adjoint method. However, the source terms of the adjoint equation are just partial derivatives of the objective functional. To make these fundamental observations for the mechanical design operational, is first to find the adjoint form of the so-called mixed problem for linear elasticity:

$$\begin{cases} \nabla \cdot \sigma + f = 0, & \text{in } \Omega; \\ \sigma(u) = \mu(\nabla u + \nabla u^T) + [\lambda(\nabla \cdot u)]\mathfrak{I} & \text{in } \Omega; \\ u = 0, & \text{on } \partial\Omega_D, \\ \sigma(u) \cdot n = g, & \text{on } \partial\Omega_N. \end{cases} \tag{1.3}$$

Here, f and g are the volume and surface loads, respectively, λ and μ are the Lamé coefficients, $\mathfrak{I}$ is the identity matrix in $\mathbb{R}^3$ and n is the outward normal on the boundary $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$.

The adjoint equation of the above state equation is simply a linear equation system containing the transpose of the system matrix (stiffness matrix) and the derivatives of the objective functional as right hand side vector. Based on the adjoint state, which is the solution of this adjoint equation, the sensitivity gradient of the probability of failure under variation of any number of the geometry nodes can be efficiently evaluated.

1.3 Sensitivity Analysis in Engineering Design

Fatigue is a structural damage process that occurs for materials subjected to cyclic loadings with stresses well below the static ultimate tensile. The failure process may prevent material components to fulfill the design requirements. This failure life can be influenced by the design geometry, material characteristics, the environment and the stress difference. In polycrystalline metal, for example, the number of load cycles until the initiation of cracks exposes a statistical scatter up to a factor 10 [22], even under lab conditions. Under repetitive or fluctuating load, the displacement of slip planes in alternating directions form small extrusions and intrusions on the material surface, which grow increasingly in depth and width, see [23, 6, 40] and Figure 1-1. This surface driven failure process is known as low cycle fatigue (LCF).
After a discretization of the linear elasticity equation, the design life, which can be understood as the number of cycle during a safe usage of the components, is derived by tacking the minimum average time of loaded points on the mesh surface with combination of safety factors used to account for the stochastic effects. The design objective, or in other words the mechanical integrity, is to maximize the lifetime.

The main goal of automated gradient based optimization in mechanical engineering is to compute effectively the design sensitivities of an objective functional with respect to some perturbations on the shape. It is common ground that these sensitivities deliver important informations to the designers. Unfortunately, the non-differentiability of the minimum mentioned previously poses a problem regarding the usefulness of such an approach by mechanical integrity. The probabilistic LCF-model introduced in [43, 44, 45, 46]

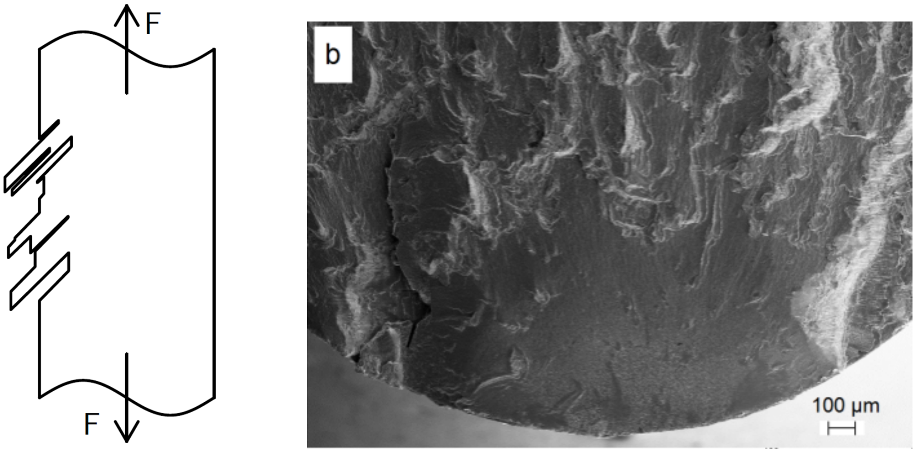


Figure 1-1: (a) Intrusions and extrusions form on the surface, (b) a crack initiation from the surface (taken from [22]).

solves this problem of differentiability by modelling the random process of LCF crack-initiation with regular objective functional expressed in term of an integral containing stress gradients informations. It is the first time that a numerical sensitivity study of such probabilistic functional is done an real life three dimensional geometries. See [8] for preliminary study in 2D.

The adjoint approach is a powerful method used to evaluate the shape sensitivity of an objective functional $J(U, X)$ with respect to design variables X in a problem governed by a system of equations represented by the residual $R(U, X) = 0$, where U is the solution of the discretized state equation.

After differentiating the objective functional with respect to design variable X, the following equation yields:

$$\frac{dJ}{dX} = \frac{\partial J}{\partial U}\frac{\partial U}{\partial X} + \frac{\partial J}{\partial X}. \tag{1.4}$$

The same for the residual yields[1]:

$$\frac{\partial R}{\partial U}\frac{\partial U}{\partial X} + \frac{\partial R}{\partial X} = 0. \tag{1.5}$$

The sensitivity can be performed by substituting the term $\frac{\partial U}{\partial X}$ in (1.4) regarding the equation (1.5). It follows:

$$\frac{dJ}{dX} = -\Lambda^T \frac{\partial R}{\partial X} + \frac{\partial J}{\partial X},$$

where the following equation

$$(\frac{\partial R}{\partial U})^T \Lambda = (\frac{\partial J}{\partial U})^T$$

is termed as the adjoint equation and Λ stands for the adjoint state.

We have to choose between two approaches for effecting the numerical sensitivity analysis: The *adjoin-then-discretize* vs. *discretize-then-adjoin* approaches. In a *adjoin-then-discretize* method, we first perform the adjoint state from the original (not discretized) partial differential equation regarding the boundary and initial conditions and then discretize the continuous adjoint equation using numerical algorithms. In a *discretize-then-adjoin* approach, one first discretize the continuous linear elasticity problem and then compute the adjoint equation to obtain a discrete sensitivity, see Figure 1-2. According to [38], the sensitivities in both approaches appear to converge to the same values as the mesh sizes go to zero. A substantial difference can be delivered by the choice of practicable mesh sizes. Since the second derivatives of the original solution are required, the use of the first approach is not possible, because these derivatives are not included in the H^1-element classes of the commercial solvers.

---

[1] Note that both equations include tensor contractions as appropriate

1.4 Outline

Concluding this introduction, we outline the structure of the remainder of the thesis. We begin chapter 2 with an introduction to basic aspects of the theory of finite element approximations (FEA). The linear isotropic elasticity model, which describes the mechanical behavior of metallic components under stress loading, are presented in section 2.1 as partial differential equation, known as the boundary value problem (BVP). The section 2.2 gives a general definition of the finite elements, presents the construction principles of a mesh and introduces the Galerkin method, which allow an approximation of the infinite function space with a finite-dimensional space, in order to solve numerically the BVP. For the purpose, to compute the surface integral in the objective functional defined in (1.2), we give quadrature formulas even with high order to take in account the nonlinearities in the integrand.

Chapter 3 introduces a probabilistic model based on fatigue failure analysis and Poisson point process. This LCF-model, developed in [46], describes the failure mechanism of surface driven LCF. After a short introduction to fatigue of materials, section 3.2 outlines the mathematical background of the probabilistic model. A discretization of the objective functional and the bilinear form with the use of the finite element method is presented in section 3.3. We apply this LCF model to mechanical components such a cantilever beam and a jet engine radial turbo compressor and investigate their probabilistic LCF life.

Chapter 4 is dedicated to the deviation of global terms for shape sensitivity analysis. In section 4.1 we review the direct and adjoint approaches designated to perform the total shape sensitivity. Then, we give details of calculations of the partial derivative terms of the sensitivity equations. Therefore, we outline the developed numerical algorithms needed for the implementation of the LCF sensitivity model. Section 4.2 provides some numerical examples and validations.

Chapter 5 is devoted to a generalization of the previous shape sensitivity analysis. Here thermal stress will be considered additionally. The linear thermoelasticity problem and the heat transfer equation are presented in section 5.1. Section 5.2 describes the coupled weak-formulation of the thermo-mechanical system. A discretization of this algebraic system is given in section 5.3. Section 5.4 is concerned with the introduction of a temperature dependent objective functional and its discretization. A shape sensitivity analysis for the thermoelasticity is presented in section 5.5.

In a final section, we summarize our conclusions and give an outlook to future research directions.

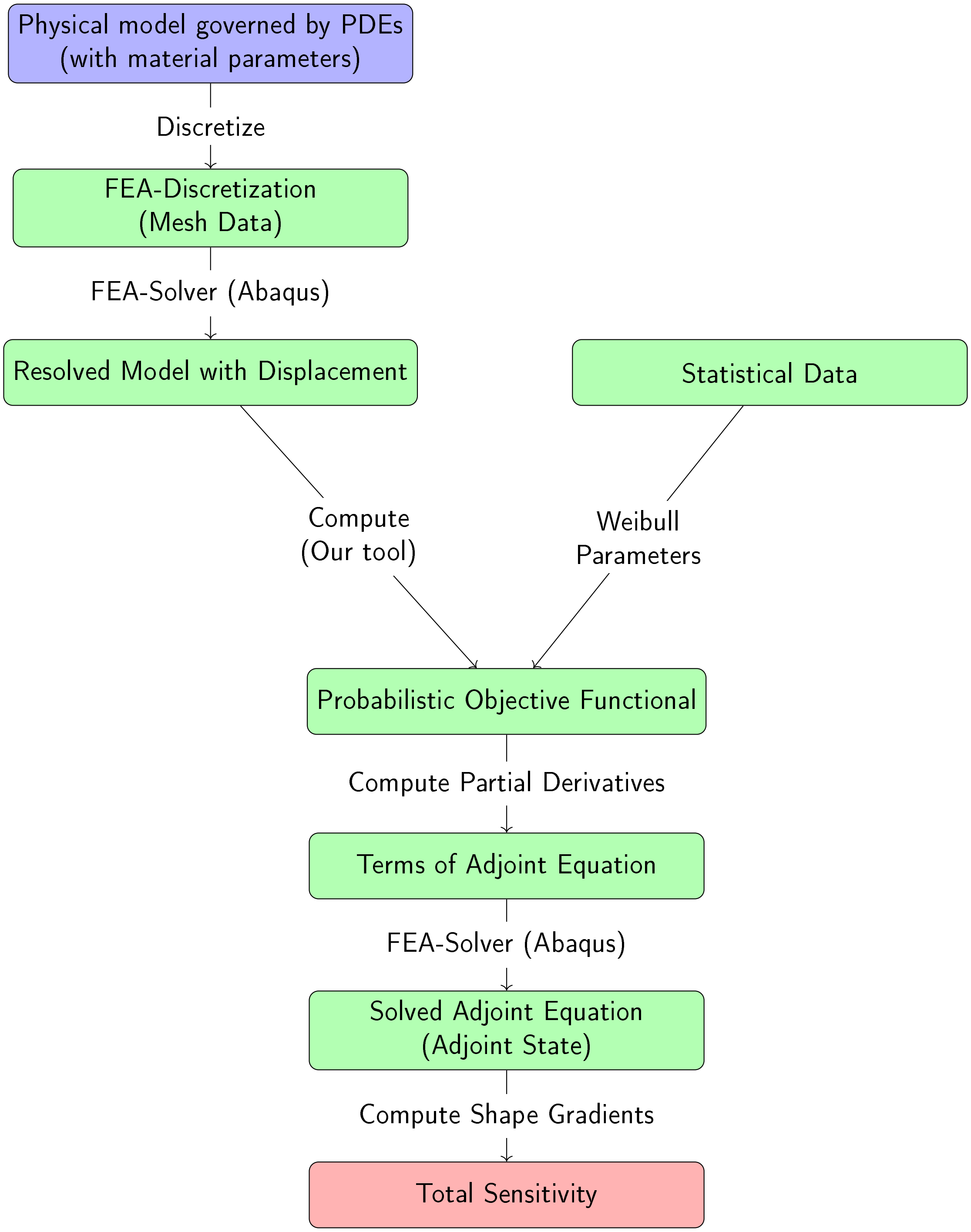


Figure 1-2: Flow diagram of shape derivative computations of the probabilistic objective functional by using adjoint method.

Chapter 2

Finite Element Approximation

Finite element methods represent powerful techniques to approximately compute solutions of partial differential equations that arise in many engineering and scientific problems. We use these techniques to solve numerically the mixed problem of linear isotropic thermoelasticity. This chapter gives an overview of the mathematical theory of finite elements [14, 15, 17].

2.1 Boundary Value Problem

Before we start with the mathematical background of the finite element approach, we describe in the next section the physical model and its governing partial differential equation.

2.1.1 Boundary Value Problem

Let us consider a domain $\Omega \subset \mathbb{R}^3$ which represents a deformable medium such as polycrystalline metal initially at equilibrium. The function $f : \Omega \to \mathbb{R}^3$ represents the external load applied to Ω. Moreover let $u : \Omega \to \mathbb{R}^3$ be the displacement field and let $\partial\Omega$ be the boundary of Ω. The clamped boundary $\partial\Omega_D$ and the Neumann boundary $\partial\Omega_N$

form a partition of $\partial\Omega$. The normal load imposed on $\partial\Omega_N$ is described by the function $g : \partial\Omega_N \to \mathbb{R}^3$. Let $\sigma : \Omega \to \mathbb{R}^{3,3}$ be the stress tensor in the medium. The model problem of linear isotropic elasticity is described by the following partial differential equation [17]:

$$\begin{cases} \nabla \cdot \sigma(u) + f = 0, & \text{in } \Omega \\ \sigma(u) = \lambda(\nabla \cdot u)\mathcal{I} + \mu(\nabla u + \nabla u^T), & \text{in } \Omega, \\ u = 0, & \text{on } \partial\Omega_D, \\ \sigma(u) \cdot n = g, & \text{on } \partial\Omega_N. \end{cases}$$

Here, λ and μ are the Lamé coefficients, $\mathcal{I}$ is the identity matrix in $\mathbb{R}^3$ and n is the outward normal on $\partial\Omega$. The linearized strain rate tensor $\varepsilon(u) : \Omega \to \mathbb{R}^{3,3}$ is defined as:

$$\varepsilon(u) = \frac{1}{2}(\nabla u + \nabla u^T).$$

This partial differential equation will be called the boundary value problem (BVP).

2.1.2 Weak Formulation

Let $v : \Omega \to \mathbb{R}^3$ be a test function and consider the functional test space $V_{DN} = \{v \in [H^1(\Omega)]^3;\ v = 0 \text{ on } \partial\Omega_D\}$ equipped with the norm $\| v \|_{1,\Omega} = \sum_{i=1}^{3} \| v_i \|_{1,\Omega}$ where $v = (v_1, v_2, v_3)^T$ and $\| v_i \|_{1,\Omega}$ is the Sobolev $W^{1,2}(\Omega)$ norm. The weak formulation of our problem is:

$$\begin{cases} \text{Find } u \in V_{DN} \text{ such that} \\ B(u, v) = L(v), \qquad \forall v \in V_{DN}, \end{cases} \tag{2.1}$$

with the bilinear form

$$B(u, v) = \lambda \int_{\Omega} \nabla \cdot u \nabla \cdot v \, dx + 2\mu \int_{\Omega} \varepsilon(u) : \varepsilon(v) \, dx.$$

and the linear form

$$L(v) = \int\limits_{\Omega} f \cdot v \, dx + \int\limits_{\partial\Omega_N} g \cdot v \, dx$$

2.2 Finite Elements

The variational abstract problem introduced in the last section is posed on a Banach space V_{DN}. An analytical solution of such problem is not always possible. The finite element approximation (FEA) can be used to obtain an approximative solution. The main idea of FEA is to divide the space V_{DN} into a number of small subdomains and to approximate the solution of the partial differential equation by a simpler polynomial function on each subdomain. This technique will be discussed in this section.

2.2.1 Definition of Finite Element

In this Section we introduce a mathematical definition of a finite element following [14, 17]:

Definition 2.2.1 ***Finite element***
A finite element is defined as a triple $\{K, P(K), \Sigma(K)\}$*, where*

- $K \subset \mathbb{R}^3$ *is a compact, connected Lipschitz set with nonempty interior called* ***element domain****,*
- $P(K)$ *is a finite-dimensional vector space of functions on* K *and*
- *the set* $\Sigma(K) = \{\varphi_1, ..., \varphi_{n_{sh}}\}$ *of linear forms* $\varphi_l : P(K) \to \mathbb{R}$ *for* $l = 1, ..., n_{sh}$ *is a basis for* $\mathcal{L}(P(K); \mathbb{R})$ *(the bounded linear functionals on P(K)).*

Note that the linear forms $\{\varphi_1, ..., \varphi_{n_{sh}}\}$ are called **the local degrees of freedom**. The basis functions $\{\theta_1, ..., \theta_{n_{sh}}\}$ in $P(K)$ which satisfies $\varphi_i(\theta_j) = \delta_{ij}$ for $1 \le i, j \le n_{sh}$ are called **local shape functions**.

We introduce a definition of an important class of finite elements which we will use in the following sections.

Definition 2.2.2 *We call* $\{K, P(K), \Sigma(K)\}$ *a* ***Lagrange finite element****, if there is a set of points* $\{X_1^K, \ldots, X_{n_{\mathrm{sh}}}^K\} \in K$ *such that, for all* $p \in P(K)$*,* $\varphi_i(p) = p(X_i^K)$*,* $1 \leq i \leq n_{sh}$*. We call* $\{X_1^K, \ldots, X_{n_{\mathrm{sh}}}^K\}$ *the nodes of the finite element.*

Definition 2.2.3 ***Local Interpolation operator***

Given a finite element $\{K, P(K), \Sigma(K)\}$*. Let* $V(K)$ *be a normed vector space of functions* $v : K \to \mathbb{R}^m$ *with* $P(K) \subset V(K)$*. We assume that the basis* $\{\varphi_1, ..., \varphi_{n_{sh}}\}$ *can be extended to* $\mathcal{L}(P(K); \mathbb{R})$*. A local interpolation operator will be defined as:*

$$\begin{aligned}\pi_K : V(K) \to & P(K) \\ v \to & \sum_{i=1}^{n_{sh}} \varphi_i(v)\theta_i\end{aligned}$$

Remark 2.2.4 *The interpolation operator* π_K *is linear and satisfies* $\varphi_i(\pi_K(v)) = \varphi_i(v)$ *for* $1 \leq i \leq n_{sh}$ *and* $\pi_K(v) = v$ *for* $v \in P(K)$*. By matching points value* v *the operator* $\pi_K(v)$ *is not necessarily defined. Thus the term interpolation can not be viewed in the classical sense.*

In general, the finite dimensional space $P(K)$ used in the definition of finite elements consists of polynomials. We give now some important spaces corresponding the finite elements like tetrahedrons and bricks.

Definition 2.2.5 ***Polynomial spaces*** $\mathbb{P}_k$ ***and*** $\mathbb{Q}_k$ *:*

Let $x = (x_1, ..., x_d)$*,* $\alpha = (\alpha_1, ..., \alpha_d)$ *and* $k \geq 0$*. The polynomial space* $\mathbb{P}_k$ *is defined as the space of polynomials of global degree at most* k *in each variables* $x_1, ..., x_d$ *with real* *coefficients* $\alpha_1, ..., \alpha_d$*,*

$$\mathbb{P}_k = span\left\{\prod_{i=1}^{d} x_i^{\alpha_i} : \alpha_i \geq 0 \ for \ i = 1, ..., d, \ \sum_{i=1}^{d} \alpha_i \leq k\right\}.$$

Similarly is the polynomial space $\mathbb{Q}_k$ *defined as:*

$$\mathbb{Q}_k = span\left\{\prod_{i=1}^{d} x_i^{\alpha_i} : \alpha_i \geq 0 \ for \ i = 1, \ldots, d, \ 0 \leq \alpha_1, \ldots, \alpha_d \leq k\right\}.$$

2.2.2 Mesh

The process of mesh generation consists of dividing the geometrical domain Ω into small elements. This decomposition will be called mesh generation. In this section we give the basic concepts of mesh construction.

Definition 2.2.6 ***Mesh***

A mesh is defined as a union of compact, connected, Lipschitz sets K_i *with* $\mathring{K}_i \neq \emptyset$ *such that* $\{K\}_{1 \leq i \leq N_{el}}$ *forms a partition of* Ω*, i.e.,*

- $\mathring{K}_i \cap \mathring{K}_j = \emptyset$ *for* $i \neq j$
- $\bigcup_{i=1}^{N_{el}} K_i = \overline{\Omega}$

$\{K\}_{1 \leq i \leq N_{el}}$ are called mesh elements. In the literature a mesh is offen denoted by $\mathcal{T}_h$. By setting $h_K = diam(K)$ for $K \in \mathcal{T}_h$, the subscript h will be defined as $h = \max\{h_K, \ K \in \{K_i\}_{1 \leq i \leq N_{el}}\}$. The parameter h can be seen as refinement indicator for a mesh.
A mesh can be generated from a fixed element $\hat{K} \subset \mathbb{R}^d$ called **reference element** and a set of geometric maps $T_K : \hat{K} \rightarrow K$, which map $\hat{K}$ to the current mesh element.
We assume that the map T_K is bijective and all mesh elements are generated from the same reference element.

Starting from the reference element $\hat{K}$ we construct a new reference finite element $(\hat{K}, \hat{P}(\hat{K}), \hat{\Sigma}(\hat{K}))$ by adding a finite-dimensional vector space $\hat{P}(\hat{K})$ and a set of linear forms $\hat{\Sigma}(\hat{K})$.

Definition 2.2.7 ***Geometric Transformation***
Given a Lagrange finite element $(\hat{K}, \hat{P}_{geo}(\hat{K}), \hat{\Sigma}_{geo}(\hat{K}))$ *with nodes* $\{\hat{X}_1^m, \dots, \hat{X}_{n_{\mathrm{geo}}}^m\}$ *and* $\{\hat{\theta}_1, ..., \hat{\theta}_{n_{geo}}\}$ *shape functions spanning the space* $\hat{P}_{geo}$ *where* $n_{geo} = card(\hat{\Sigma}_{geo})$ *and* $1 \leq m \leq N_{el}$. *Then the geometric transformation is defined as:*

$$\begin{aligned} T_m &: \hat{K} \to \mathbb{R}^d \\ \hat{x} &\to T_m(\hat{x}) = \sum_{i=1}^{n_{geo}} \hat{X}_i^m \hat{\theta}_i(\hat{x}) \end{aligned}$$

To ensure that the transformation T_m is a diffeomorphism, the numbering of the reference nodes $\{\hat{X}_1^m, \dots, \hat{X}_{n_{\mathrm{geo}}}^m\}$ and the nodes $\{X_1^m, \dots, X_{n_{\mathrm{geo}}}^m\}$ have to be compatible. Note that the compatibility impose that the numbering of nodes is such that the Jacobian determinant of the transformation T_m is positive.
By mesh generation a list

$$X_i^m, \ 1 \leq i \leq n_{geo}, \ 1 \leq m \leq N_{el}$$

will be created. The triple $(\hat{K}, \hat{P}_{geo}(\hat{K}), \hat{\Sigma}_{geo}(\hat{K}))$ is called the **geometric reference finite element**, $\{\hat{\theta}_1, ..., \hat{\theta}_{n_{geo}}\}$ the **geometric reference shape functions** and $\{\hat{X}_1^m, \dots, \hat{X}_{n_{\mathrm{geo}}}^m\}$ the **geometric reference nodes**.

Definition 2.2.8 ***Affine meshes***
A mesh is said to be affine if all transformations $\{T_m\}_{1 \leq m \leq N_{el}}$ *are affine i.e.*

$$T_m : \widehat{K} \mapsto K, \hat{x} \mapsto J_K \hat{x} + a_K$$

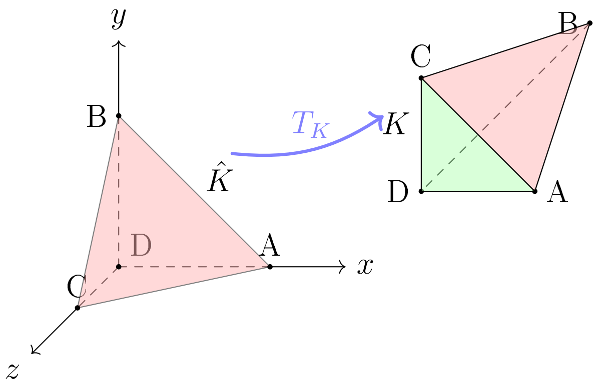


Figure 2-1: Geometric transformation of tetrahedral element

where $J_K \in \mathbb{R}^{d,d}$ *and* $a_K \in \mathbb{R}^d$.

From a practical point of view, the use of geometric transformations reduces the calculation of integral over the whole domain to evaluate integrals over the reference element $\widehat{K}$. Note that the use of affine meshes produces an interpolation error by domains with curved boundaries.

In order to approximate the domain Ω, we consider the following set :

$$\overline{\Omega}_h = \bigcup_{K \in \mathcal{T}_h} K.$$

The open set Ω_h is defined as a **geometrical interpolation** of the domain Ω.

Let $K = T_K(\hat{K})$ be a mesh element. Vertices, edges and faces of K are the image under the transformation T_K of vertices, edges and faces of the reference element $\hat{K}$.

Definition 2.2.9 ***Conformal meshes***

Let $\Omega \subset \mathbb{R}^d$ *be a domain and let* $\mathcal{T}_h = \{K_i\}_{1 \le i \le N_{el}}$ *be a mesh of* Ω *. A mesh* $\mathcal{T}_h$ *is called geometrically conformal if for all* K_i *and* K_j *with* $F = K_i \cap K_j \neq \emptyset$ $(i \neq j)$ *there is a face* $\widehat{F}$ *of* $\widehat{K}$ *such that*

$$F = T_i(\widehat{F}) = T_j(\widehat{F}) \text{ and } T_{i|\widehat{F}} = T_{j|\widehat{F}}.$$

2.2.3 Construction of Finite Elements

In this section we present the technique used to generate finite elements based on a fixed reference element. Let $\{K, P(K), \Sigma(K)\}$ be a finite element and let $\{\hat{K}, \hat{P}(\hat{K}), \hat{\Sigma}(\hat{K})\}$ be a fixed finite element. Moreover let $V(K)$ and $V(\hat{K})$ two Banach spaces of functions with values in $\mathbb{R}^m$. For every element $K \in \mathcal{T}_h$ let

$$\psi_K : V(K) \to V(\hat{K})$$

be a linear bijective mapping. The **local interpolation operator** $\pi_{\hat{K}}$ is defined as:

$$\begin{aligned} \pi_{\hat{K}} : V(\hat{K}) \to & \hat{P}(\hat{K}) \\ \hat{v} \to & \sum_{i=1}^{n_{sh}} \hat{\varphi}_i(\hat{v}) \hat{\theta}_i \end{aligned}$$

Proposition 2.2.10 ***Generation of finite elements***
Let $\mathcal{T}_h$ be a mesh and let $\{\hat{K}, \hat{P}(\hat{K}), \hat{\Sigma}(\hat{K})\}$ be a fixed finite element. For $K \in \mathcal{T}_h$ the triplet $\{K, P(K), \Sigma(K)\}$ so that:

$$\begin{cases} K = T_K(\hat{K}) \\ P(K) = \psi_K^{-1}(\hat{P}) \\ \Sigma(K) = \{\hat{\varphi}_i(\psi_K(P_K)),\ 1 \le i \le n_{sh}\} \end{cases}$$

is a finite element.

The finite element $\{\hat{K}, \hat{P}(\hat{K}), \hat{\Sigma}(\hat{K})\}$ used to generate finite elements in the above proposition will be called **reference finite element** .
By setting $\varphi_{K,i}(p) = \hat{\varphi}_i(\psi_K(p))$ for $1 \le i \le n_{sh}$ and $\forall p \in P_K$, we define the **local shape**

functions as $\theta_{K,i} = \psi_K^{-1}(\hat{\theta}_i)$ and the **associated local interpolation operator** as:

$$\begin{aligned} \pi_K : V(K) \to& P(K) \\ v \to& \sum_{i=1}^{n_{sh}} \varphi_{K,i}(v)\theta_{K,i} \end{aligned}$$

Consequently, the following diagram commutes:

$$\begin{tikzcd} V(K) \arrow[r, "\psi_K"] \arrow[d, "\pi_K"] & V(\hat{K}) \arrow[d, "\pi_{\hat{K}}"] \\ P(K) \arrow[r, "\psi_K"] & \hat{P}(\hat{K}). \end{tikzcd}$$

Definition 2.2.11 ***Isoparametric***

Given a reference finite element $\{\hat{K}, \hat{P}(\hat{K}), \hat{\Sigma}(\hat{K})\}$ *and a geometric reference finite element* $\{\hat{K}, \hat{P}_g(\hat{K}), \hat{\Sigma}_g(\hat{K})\}$. *The interpolation is called isoparametric when* $\{\hat{K}, \hat{P}(\hat{K}), \hat{\Sigma}(\hat{K})\}$ *and* $\{\hat{K}, \hat{P}_g(\hat{K}), \hat{\Sigma}_g(\hat{K})\}$ *are identical.*

The interpolation is called subparametric if $\hat{P}_g(\hat{K}) \subsetneq \hat{P}(\hat{K})$. For examples see figure 2-2.

Based on the local interpolation operator defined previously, we construct in the next a global interpolation operator as follow:
Let Ω_h be a geometric interpolation of Ω. we choose a domain $D(\pi_h)$

$$D(\pi_h) = \{v \in [L^1(\Omega_h)]^m, \forall K \in \mathfrak{T}_h, v|_K \in V(K)\}.$$

For a fixed element $K \in \mathfrak{T}_h$ we define elementwise

$$(\pi_h v)|_K = \pi_K(v|_K) = \sum_{i=1}^{n_{sh}} \varphi_{K,i}(v|_K)\theta_{K,i}$$

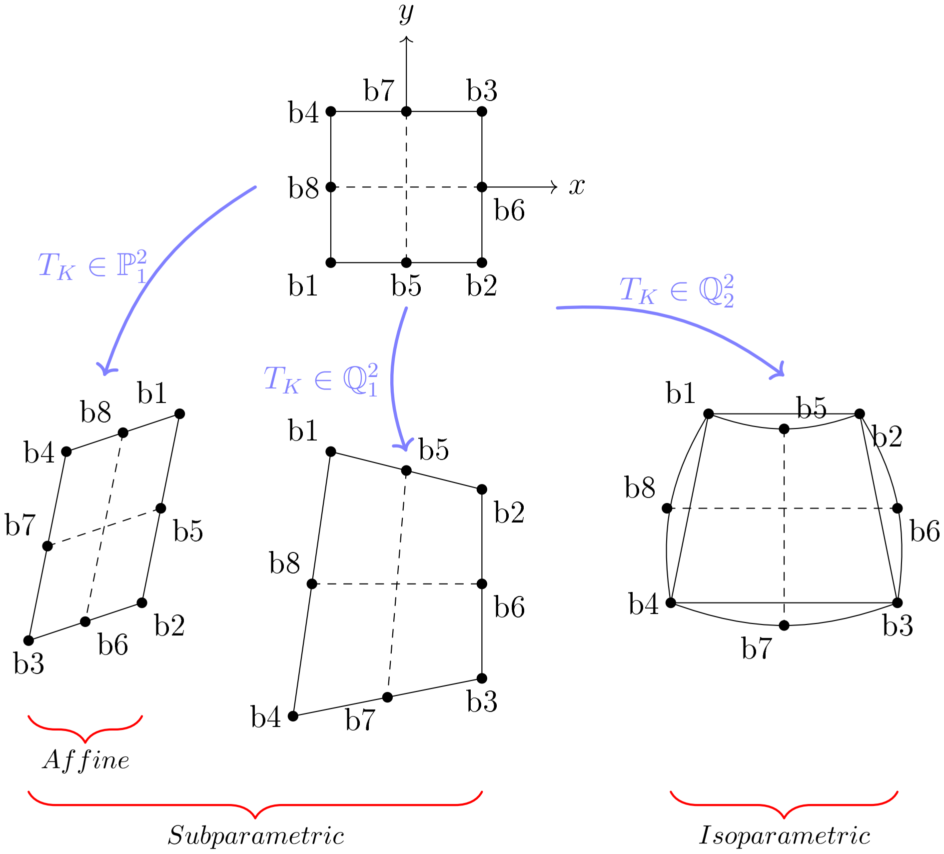


Figure 2-2: Affine, Subparametric and Isoparametric transformation [15]

Now we define the **global interpolation operator** over the domain $D(\pi_h)$ as:

$$\begin{aligned} \pi_h : D(\pi_h) &\rightarrow W_h \\ v &\rightarrow \sum_{K \in \mathfrak{T}_h} \sum_{i=1}^{n_{sh}} \varphi_{K,i}(v|_K)\theta_{K,i} \end{aligned}$$

where W_h is defined as

$$W_h = \{v_h \in [L^1(\Omega_h)]^m, \forall K \in \mathfrak{T}_h, v|_K \in P(K)\}.$$

Definition 2.2.12 ***approximation space***

Let Ω_h *a geometric interpolation of* Ω *i.e.* $(\overline{\Omega}_h = \bigcup_{K \in \mathcal{T}_h} K)$. *The space* W_h *is called an approximation space.*

Definition 2.2.13 ***Conformal approximation***

Let W_h *an approximation space and let* V *be a Banach space. If* $W_h \subset V$ *holds, the space* W_h *is said to be* V*-conformal.*

2.2.4 Construction of H^1−conformal Subspace

Before we start with the construction of H^1-conformal subspace of the approximation space W_h, let us give some technical definitions that we need.

Let F^i be a $(d-1)-$manifold and we suppose that there are $K_1,\ K_2 \in \mathcal{T}_h$ such that $F^i = K_1 \cap K_2$. The face F^i is called a **interior face** and the set of interior faces will be denoted by $\mathcal{F}_h^i$. The set $\mathcal{F}_h^\partial$ of faces F^s separating the mesh from the exterior of Ω_h is defined as a set of $(d-1)-$manifolds satisfying the condition that there is $K \in \mathcal{T}_h$ such that $F^s = K \cap \partial\Omega_h$. We set $\mathcal{F}_h = \mathcal{F}_h^i \cup \mathcal{F}_h^\partial$.

Let $F^i = K_1 \cap K_2$ and let n_1 and n_2 denote the outward normal to K_1 and K_2 respectively. On every finite element $K \in \mathcal{T}_h$ we define a scalar-valued function v and suppose that v is sufficiently smooth to have limits on both sides of F. Then the **jump** of v across F is defined as:

$$\|v\|_F = v|_{K_1} n_1 + v|_{K_2} n_2.$$

Let the reference finite element $\{\hat{K}, \hat{P}(K), \hat{\Sigma}(K)\}$ be a Lagrange finite element and let the mesh $\mathcal{T}_h$ be geometrically conformal. By setting

$$\begin{cases} V(\hat{K}) = [\mathcal{C}^0(\hat{K})]^m, \\ V(K) = [\mathcal{C}^0(K)]^m, \\ \psi_K(v) = v \circ T_K \in V(\hat{K}),\ \forall v \in V(K) \end{cases}$$

the proposition 2.2.10 guarantees that the finite element $\{K, P(K), \Sigma(K)\}$ is a Lagrange finite element for every $K \in \mathcal{T}_h$.

Proposition 2.2.14 *We consider the following approximation space with zero jumps:*

$$V_h = \{v_h \in W_h;\ \forall F \in \mathcal{F}_h^i,\ \|v_h\|_F = 0\}.$$

Then we have:

$$V_h \subset [H^1(\Omega_h)]^m.$$

Given the local degrees of freedom of adjacent elements. Under which properties is the zero jump condition holds?. We consider the following conditions:

1. For each face $\hat{F}$ of the reference element $\hat{K}$, the number of nodes $n_{n_{sh}}^{\partial}$ is the same.
2. The triplet $\{\hat{F}, \hat{P}(\hat{F}), \hat{\Sigma}(\hat{F}\})$ is a finite element, where $\hat{F}$ is a face of $\hat{K}$ with nodes $\{X_{1,\hat{F}}, ...X_{n_{n_{sh}}^{\partial},\hat{F}}\}$, $\hat{P}(\hat{F}) = \{\hat{q}; \exists\hat{p} \in \hat{P}, \hat{q} = \hat{P}|_{\hat{F}}\}$ and $\Sigma(\hat{F}) = \{\hat{\varphi}_1, ..., \hat{\varphi}_{n_{n_{sh}}^{\partial}}\}$ such that $\hat{\varphi}_i(\hat{q}) = \hat{q}(X_{i,\hat{F}})$ for $\hat{q} \in \hat{P}(\hat{F})$ and $1 \leq i \leq n_{n_{sh}}^{\partial}$.
3. $\forall F^i \in \mathcal{F}_h^i$ with $F = K_s \cap K_t,\ s \neq t$, assume that there are renumberings of the Lagrange nodes K_s and K_t so that $X_{K_s,i} = X_{K_t,i}$ for $i \in \{1, ..., n_{n_{sh}}^{\partial}\}$.

We consider the three conditions mentioned above and let $u_h \in W_h$. Then the following properties are equivalent [17]:

- $\forall F^i \in \mathcal{F}_h^i, ||u_h||_{F^i} = 0$
- $\forall F^i \in \mathcal{F}_h^i$ such that $F^i = K_s \cap K_t (s \neq t)$,

$$\forall i \in \{1, ..., n_{n_{sh}}^{\partial}\}, u_{h|K_s}(X_{K_s,i}) = u_{h|K_t}(X_{K_t,i}).$$

We define $\{X_1, ..., X_N\} = \bigcup_{K\in\mathfrak{T}_h}\{X_1^K, ..., X_{n_{sh}}^K\}$ as the set of all the Lagrange nodes. For $K \in \mathfrak{T}_h$ and $m \in \{1, ..., n_{sh}\}$, let

$$\begin{aligned}\widehat{j}:\ \mathfrak{T}_h \times \{1,\dots,n_{\mathrm{sh}}\} \to& \{1,\dots,N\}\\ (K,m) \to& \widehat{j}(K,m)\end{aligned}$$

be the mapping which allow us to establish the relation between local and global index of nodes.

Let $\{\phi_1, ..., \phi_N\}$ be a set of functions in W_h defined as:

$$\phi_{i|K}(X_m^K) = \begin{cases} \delta_{mn}, & \text{if there is } n \in \{1, ..., n_{sh}\} \text{ such that } i = \widehat{j}(K,n);\\ 0, & \text{otherwise.}\end{cases}$$

This implies that $\phi_i(X_j) = \delta_{ij}$ for $1 \leq i, j \leq N$.

According to [17, Prop. 1.78] the set of functions $\{\phi_1, ..., \phi_N\}$ is a basis in V_h and called the **global shape functions**. Moreover we define for $1 \leq i \leq N$ the following linear forms:

$$\begin{aligned}\gamma_i : V_h \to& \mathbb{R}\\ v_h \to& \gamma_i(v_h) = v_h(X_i).\end{aligned}$$

The set $\{\gamma_1, ..., \gamma_N\}$ is a basis in $\mathcal{L}[V_h, \mathbb{R}]$ and the linear forms are called the **global degrees of freedom** in V_h.

Now, we define the global Lagrange interpolation operator as:

$$\begin{aligned} \pi_h : \mathcal{C}^0(\bar{\Omega}_h) &\rightarrow V_h \\ v &\rightarrow \pi_h(v) = \sum_{i=1}^{N} v(X_i)\phi_i \end{aligned}$$

As examples for approximation space we give:

$$P^k_{c,h} = \{v_h \in \mathcal{C}^0(\bar{\Omega}_h); \forall K \in \mathcal{T}_h, v_h \circ T_K \in \mathbb{P}_k\} \tag{2.2}$$

and

$$Q^k_{c,h} = \{v_h \in \mathcal{C}^0(\bar{\Omega}_h); \forall K \in \mathcal{T}_h, v_h \circ T_K \in \mathbb{Q}_k\} \tag{2.3}$$

2.2.5 Galerkin Approximation

In this section we consider a linear problem such (2.1) and we will see under which conditions this problem is said to be well-posed. Then, we introduce the idea of Galerkin approximation technique.

Well-posedness

Let W be a Banach space equipped with the norm $||.||_W$ and let V be a reflexive Banach space with the norm $||.||_V$. Moreover we consider a continuous bilinear form B on $W \times V$ and a continuous linear form f on V. The following problem

$$\begin{cases} \text{Find } u \in W \text{ such that} \\ B(u,v) = f(v), \qquad \forall v \in V, \end{cases} \tag{2.4}$$

is well-posed (in the sense of Hadamard) if it admits a unique solution and if the following holds:

$$\exists c > 0, \forall f \in \mathcal{L}(V,\mathbb{R}), ||u|| \leq c||f||_{\mathcal{L}(V,\mathbb{R})}.$$

The next lemma gives a sufficient condition for existence and uniqueness of a solution of Problem 2.4.

Lemma 2.2.15 ***Lax-Milgram***

Let V be a Hilbert space and let B be a continuous bilinear form on $V \times V$. We assume that B is coercive, i.e. $\exists \alpha > 0, \forall u \in V B(u,u) > \alpha ||u||_V^2$. Moreover let f be a continuous linear form on V. Then, the following problem

$$\begin{cases} \textit{Find } u \in V \textit{ such that} \\ B(u,v) = f(v), \qquad \forall v \in V, \end{cases}$$

is well-posed and

$$\forall f \in \mathcal{L}(V, \mathbb{R}), ||u||_V \leq \frac{1}{\alpha} ||f||_{\mathcal{L}(V,\mathbb{R})}.$$

Proof. See [17]. ■

The coercivity of the continuous bilinear form B results from the following Korn inequalities:

Theorem 2.2.16 ***(Korn's first inequality)***

Let $\Omega \subseteq \mathbb{R}^3$. Then, there exists c such that

$$\forall u \in [H_0^1(\Omega)]^3, c\|u\|_{1,\Omega} \leq \left(\int_\Omega \varepsilon(u) : \varepsilon(u) \, dx \right)^{1/2}.$$

Proof. See [17]. ■

Theorem 2.2.17 ***(Korn's second inequality)***

Let $\Omega \subseteq \mathbb{R}^3$. Then, there exists c such that

$$\forall u \in [H^1(\Omega)]^3, c\|u\|_{1,\Omega} \leq \left(\int_\Omega \varepsilon(u) : \varepsilon(u) \, dx \right)^{1/2} + \left(\int_\Omega u^2 \, dx \right)^{1/2}.$$

Proof. See [13]. ■

Since the coercivity holds only on Hilbert space, we cannot apply the Lax-Milgram lemma to Banach spaces. The theorem of Banach-Necas-Babuska generalize the lemma of Lax-Milgram to Banach spaces by giving equivalent conditions for the existence and uniqueness of solution of the Problem 2.4.

Theorem 2.2.18 ***Banach-Necas-Babuska***
Let W be a Banach space and V be a reflexive Banach space, $B \in \mathcal{L}(W \times V, \mathbb{R})$ and $f \in \mathcal{L}(V, \mathbb{R})$. Then problem 2.4 is well posed if and only if:

- $\exists \alpha > 0, \inf_{w \in W} \sup_{v \in V} \frac{B(w,v)}{||w||_W ||v||_V} \geq \alpha,$
- $\forall v \in V, (\forall w \in W, B(w,v) = 0) \Rightarrow (v = 0).$

Moreover,

$$\forall f \in \mathcal{L}(V, \mathbb{R}), ||u||_W \leq \frac{1}{\alpha} ||f||_{\mathcal{L}(V,\mathbb{R})}.$$

Proof. We refer to [17] page 85. ■

2.2.6 Galerkin Methods

The central idea of Galerkin approximation is to seek the solution of the problem (2.4) not in the infinite dimensional spaces W and V but in the finite-dimensional spaces W_h and V_h. We have seen in the section 2.2 the interpolation technique used to construct such spaces W_h and V_h where the index h refers to the mesh size. By replacing both spaces by its approximations in the problem (2.4) we have now to solve the following problem:

$$\begin{cases} \text{Find } u \in W_h \text{ such that} \\ B(u,v) = f(v), \qquad \forall v \in V_h, \end{cases} \tag{2.5}$$

Let us consider that $\dim(W_h) = N$ and $\dim(V_h) = M$. The family of functions $\{\theta_1, ..., \theta_N\}$ is a basis of W_h and the family of functions $\{\phi_1, ..., \phi_M\}$ is a basis of V_h. For each $u_h \in W_h$ we can write:

$$u_h = \sum_{i=1}^{N} U_i \theta_i$$

where $U_i \in \mathbb{R}$ for $1 \leq i \leq M$. Setting in (2.4) becomes

$$B(\sum_{j=1}^{N} U_j \theta_j, \phi_i) = f(\phi_i) \text{ for } i = 1, ..., M$$

and with use of linearity of B

$$\sum_{j=1}^{N} U_j B(\theta_j, \phi_i) = f(\phi_i) \text{ for } i = 1, ..., M.$$

We put

$$B_{ij} = B_h(\theta_j, \phi_i), \text{ for } 1 \leq i \leq M, 1 \leq j \leq N$$

and

$$F_i = f(\theta_i), \ 1 \leq i \leq M$$

Finally we have the linear system

$$BU = F.$$

The matrix $B \in \mathbb{R}^{N \times M}$ with entries B_{ij} will be called **stiffness matrix** and the vector $F = (F_i)_{1 \leq i \leq M}$ will be called **force** vector.

H^1-conformal Approximation

Let $(\hat{K}, \hat{P}(\hat{K}), \hat{\Sigma}(\hat{K}))$ be a Lagrangian reference finite element and we consider the approximation spaces $P^k_{c,h}$ and $Q^k_{c,h}$ defined on (2.2) and (2.3) respectively. To construct a $V-$conformal approximation space, we must take into account the boundary conditions.

Thus we set

$$V_h^p = P_{c,h}^k \cap V \text{ and } V_h^q = Q_{c,h}^k \cap V.$$

In our problem we have a mixed boundary conditions: the Dirichlet boundary $\partial\Omega_D$ and the Neumann boundary $\partial\Omega_N$. By assuming that $\partial\Omega_D$ is a collection of faces we can write

$$V_h^p = \{v_h \in P_{c,h}^k; v_h = 0 \text{ on } \partial\Omega_D\} \text{ and } V_h^q = \{v_h \in Q_{c,h}^k; v_h = 0 \text{ on } \partial\Omega_D\}$$

as H^1-conformal approximation spaces.

2.2.7 Numerical Integration

In this section we introduce the most common integration technique used by the finite element method for approximating integrals over a given function. We give firstly a definition of quadrature [17].

Definition 2.2.19 ***Quadrature***
We consider a non-empty, Lipschitz, compact, connected subset $K \subset \mathbb{R}^d$. A quadrature on K with l_q points $l_q \geq 1$ is defined as a union $\{\omega_i, \xi_i\}_{i=1}^{l_q}$ of a set of l_q real numbers $\{\omega_1, \dots, \omega_{l_q}\}$ and a set of l_q points $\{\xi_1, \dots, \xi_{lq}\}$ in K.
The order k of the quadrature is the largest integer k such that:

$$\int_K p(x)\,dx = \sum_{l=1}^{l_q} \omega_l p(\xi_l), \ \forall p \in \mathbb{P}_k.$$

The set $\{\omega_1, \dots, \omega_{l_q}\}$ is called quadrature weights and $\{\xi_1, \dots, \xi_{lq}\}$ is called Gauß points or quadrature nodes.

There are various ways to approximate integrals of a function over a given domain. We use in this thesis Gauß quadratures which provide exact approximations for polynomials

l_q	k_q	Nodes	Weights
1	1	0	2
2	3	$\pm\frac{\sqrt{3}}{3}$	1
3	5	$\pm\sqrt{\frac{3}{5}}$ 0	$\frac{5}{9}$ $\frac{8}{9}$
4	7	$\pm\sqrt{\frac{1}{35}(15+2\sqrt{30})}$ $\pm\sqrt{\frac{1}{35}(15-2\sqrt{30})}$	$\frac{1}{2}-\frac{1}{6}\sqrt{\frac{5}{6}}$ $\frac{1}{2}+\frac{1}{6}\sqrt{\frac{5}{6}}$

Table 2.1: Nodes and weights for quadratures on the interval $[-1, 1]$

with degrees less than $2l_q - 1$ by an appropriate choice of quadrature coefficients and points.

Proposition 2.2.20 *For* $l_q \geq 1$ *let* $\mathcal{E}_k(x)$ *be define the Legendre polynomials on the interval* $[0, 1]$ *as:*

$$\mathcal{E}_k(x) = \frac{1}{k!}\frac{d^k}{dx^k}(x^2 - x)^k, \text{ for } k \geq 0.$$

Denote by $\xi_1, ..., \xi_{l_q}$ *the* l_q *roots of* $\mathcal{E}_k(x)$*. If we set*

$$\omega_i = \int_0^1 \prod_{j=1, j\neq i}^{l_q} \frac{x - \xi_j}{\xi_i - \xi_j} dx$$

the family $\{\omega_1, \ldots, \omega_{l_q}, \xi_1, \ldots, \xi_{lq}\}$ *is a quadrature of order* $2l_q - 1$ *on* $[0, 1]$.

For Computing integrals over a interval $[a, b]$, we can benefit from the quadratures in the last proposition 2.2.20 by using a simple change of variables. We can deduce quadratures on rectangles und bricks by subdividing the multidimensional domain into one-dimensional intervals and using Fubini's theorem.

We define the discretized finite element space $V_h^1(\Omega, \mathbb{R})$ as :

$$V_h^1(\Omega, \mathbb{R}) = span\{\varphi_j, j \in \{1, \ldots, N\}\}$$

We put $V_h^1(\Omega, \mathbb{R}^3) = H_h^1(\Omega, \mathbb{R})^{\times 3}$ and we have $V_h^1(\Omega, \mathbb{R}^3) \subseteq V^1(\Omega, \mathbb{R}^3)$. We define $V_{D,h}^1(\Omega, \mathbb{R}^3)$ as :

$$V_{D,h}^1(\Omega, \mathbb{R}^3) = \{u \in V_h^1(\Omega, \mathbb{R}^3); u = 0 \text{ on } \overline{\partial\Omega_D} \cap \{X_1, \ldots, X_n\}\}.$$

By assuming that u vanishes along the Dirichlet boundary $\partial\Omega_D$ if u vanishes on all Dirichlet nodes, we can write:

$$V_{D,h}^1(\Omega, \mathbb{R}^3) = V_h^1(\Omega, \mathbb{R}^3) \cap V_D^1(\Omega, \mathbb{R}^3).$$

We can now write the discretized elasticity problem of the problem (2.1) as:

$$\begin{cases} \text{Seek } u \in V_{D,h}^1(\Omega, \mathbb{R}^3) \text{ such that} \\ \\ B(u, v) = \int\limits_{\Omega} f \cdot v \, dx + \int\limits_{\partial\Omega_N} g \cdot v \, dA, \forall v \in V_{D,h}^1(\Omega, \mathbb{R}^3) \end{cases} \tag{2.6}$$

The coercivity of the bilinear form $B(u, v)$ on $V_{D,h}^1(\Omega, \mathbb{R}^3)$ is established by the Korn's inequalities mentioned in 2.2.16 and 2.2.17. This ensure the existence of a solution of (2.6).

Proposition 2.2.21 *Let u solve (2.1) and let u_h solve the discretized problem (2.6). Then*

$$\lim_{h \to 0} \|u - u_h\|_{[H^1(\Omega)]^3} = 0.$$

Furthermore, if $u \in [H^{l+1}(\Omega)]^3 \cap V_{DN}$ for some $l \in \{1, ..., k\}$, there exists a constant $C \geq 0$ such that

$$\|u - u_h\|_{[H^1(\Omega)]^3} \leq Ch^l |u|_{[H^{l+1}(\Omega)]^3}.$$

Proof. A direct consequence of Céa's Lemma and Corrollary 1.109 in [17]. ■

Chapter 3

Materials and Fatigue

Polycrystalline metals are composed of many small single crystals called grains which are variously oriented and sized. This microstructure plays an important role in determining the material properties such as electrical and thermal conductivity, strength and malleability [6].

The grain boundary is defined as the space that exists between grains and represents a defect in polycrystalline materials. The cooling rate has a profound effect on the size of the grain. If the cooling process is slow, large grains will be produced and they are easier to break or fracture [57].

3.1 Fatigue of Materials

In materials science, fatigue defined the damage that occurs when a material is subjected to a repetitive or fluctuating stress. Material fatigue cracks initiate and propagate in contact regions where the strain is most severe. Many material engineers are interesting to analyze the fatigue life N_{det} which is defined as the number of stress cycles to crack initiation or failure. In this work, we study the sensitivity of the failure probability of a gas turbine blade subjected to low-cycle fatigue (LCF) loding.

Experiments show that the strain amplitude can be decomposed into an elastic and plastic part as follow:

$$\varepsilon_a^{el-pl} = \varepsilon_a^{el} + \varepsilon_a^{pl}$$

High cycle fatigue (HCF) can only be achieved by low amplitude high frequency elastic strains, where only little plastic deformation occurs. In this range the total strain corresponds approximately to the elastic part and can be expressed in terms of true elastic strain amplitude as [40, 46]:

$$\varepsilon_a^{el} = \frac{\sigma_f^{'}}{E}(2N_{det})^b$$

where the parameters $\sigma_f^{'}$, b and E are called fatigue strength, fatigue strength exponent and Young's modulus, respectively.

High amplitude low frequency plastic strains characterize the low cycle fatigue (LCF). In this case, large plastic deformation occurs so that the total strain is mainly determined by the plastic strain. The Coffin-Manson equation describes the LCF range as follow:

$$\varepsilon_a^{pl} = \varepsilon_f^{'}(2N_{det})^c$$

where $\varepsilon_f^{'}$ is fatigue ductility and c is fatigue ductility exponent. Combining the Basquin equation and the Coffin-Manson equation leads to the so called Coffin-Manson-Basquin (CMB) equation:

$$\varepsilon_a^{el-pl} = \frac{\sigma_f^{'}}{E}(2N_{det})^b + \varepsilon_f^{'}(2N_{det})^c \tag{3.1}$$

From the last equation we define the CMB function as:

$$CMB : \mathbb{R}^+ \mapsto \mathbb{R},\ CMB(x) = \frac{\sigma_f^{'}}{E}(2x)^b + \varepsilon_f^{'}(2x)^c.$$

The Ramberg–Osgood equation has been proposed to describe the non linear relationship between stress and strain of many materials. This equation is especially useful for strain hardening metals with smooth transition between elastic and plastic deformation. Introducing the parameters strain hardening coefficient K and strain hardening exponent n the Ramberg-Osgood equation is given by:

$$\varepsilon_a^{el-pl} = \frac{\sigma_a^{el-pl}}{E} + \Big(\frac{\sigma_a^{el-pl}}{K}\Big)^{1/n} \tag{3.2}$$

with Young' modulus $E = \frac{\mu(3\lambda+2\mu)}{\lambda+\mu}$. We define the Ramberg-Osgood function as

$$RO : \mathbb{R}^+ \mapsto \mathbb{R},\ RO(x) = \frac{x}{E} + \Big(\frac{x}{K}\Big)^{1/n}.$$

We present the method of Neuber shakedown, which allows a conversion of elastic stress values obtained from a finite element analysis to elastic-plastic values. From the Ramberg-Osgood equation and the Neuber shakedown method we can associate the strain ε_a^{el-pl} to the purely elastic stress σ_a^e with the following composition:

$$\frac{(\sigma_a^e)^2}{E} = \sigma_a^e \varepsilon_a^e = \frac{(\sigma_a^{el-pl})^2}{E} + \sigma_a^{el-pl}\left(\frac{\sigma_a^{el-pl}}{K}\right)^{1/n}$$

By defining the shakedown function SD as:

$$SD : \mathbb{R}^+ \mapsto \mathbb{R},\ SD(x) := \frac{x^2}{E} + x\left(\frac{x}{K}\right)^{1/n},$$

we can write:

$$N_{det} = CMB^{-1} \circ RO \circ SD^{-1}\left(\frac{(\sigma_a^e)^2}{E}\right). \tag{3.3}$$

By the analysis of plastic deformations of ductile materials such a metal, it is very important to characterize the threshold between elastic and plastic deformations. Such threshold can be expressed as the so-called yield criteria. We use in our work the von Mises yield criterion which is given by:

$$\sqrt{\frac{1}{6}[(\sigma_1-\sigma_2)^2+(\sigma_1-\sigma_3)^2+(\sigma_2-\sigma_3)^2]} = k_F$$

where $\sigma_1, \sigma_2, \sigma_3$ are the principal stresses and k_F is the critical value of the yield criterion. We introduce the definition of the stress deviator tensor in $\mathbb{R}^3$:

$$\sigma_a^{dev} = \sigma_a^e - \frac{1}{3}\Big(\operatorname{tr}\sigma_a^e\Big)\mathfrak{I}.$$

The von Mises stress is defined as:

$$\begin{aligned}\sigma_\nu^{el-pl} &= \sqrt{\frac{3}{2}\operatorname{tr}\Big((\sigma^{dev})^2\Big)} \qquad (3.4)\\ &= \sqrt{\frac{1}{2}[(\sigma_1-\sigma_2)^2+(\sigma_1-\sigma_3)^2+(\sigma_2-\sigma_3)^2]}.\end{aligned}$$

Remember that $\sigma_a^e(u) = \lambda\operatorname{tr}\left(\frac{1}{2}(\nabla u+\nabla u^T)\right)\cdot\mathfrak{I} + \mu(\nabla u+\nabla u^T)$. We first consider the strain deviator tensor σ_a^{dev}:

$$\begin{aligned}\sigma_a^{dev} =& \lambda(\nabla\cdot u)\cdot\mathfrak{I} + \mu(\nabla u+\nabla u^T) - \frac{1}{3}\big(3\lambda(\sum_{i=1}^{3}\frac{\partial u_i}{\partial x_i}) + 2\mu(\sum_{i=1}^{3}\frac{\partial u_i}{\partial x_i})\big)\cdot\mathfrak{I}\\ =& \lambda(\nabla\cdot u)\cdot\mathfrak{I} + \mu(\nabla u+\nabla u^T) - \frac{1}{3}(3\lambda+2\mu)(\nabla\cdot u)\cdot\mathfrak{I}\\ =& -\frac{2}{3}\mu(\nabla\cdot u)\cdot\mathfrak{I} + \mu(\nabla u+\nabla u^T)\\ =& -\frac{2}{3}\mu\operatorname{tr}(\frac{1}{2}(q+q^T)).\mathfrak{I} + \mu(q+q^T) \text{ with } q=\nabla u.\end{aligned}$$

By substitution of the stress deviator in (3.4) we get:

$$\sigma_\nu^{el-pl} = \sqrt{\frac{3}{2}\sum_{i,j=1}^{3}\Big(-\frac{2}{3}\mu\big(\frac{1}{2}(q_{ij}+q_{ji})\big)\delta_{ij}+\mu(q_{ij}+q_{ji})\Big)^2}$$

Now we can write the term $\frac{(\sigma_\nu^{el-pl})^2}{E}$ as:

$$\begin{aligned}\frac{(\sigma_\nu^{el-pl})^2}{E} &= \frac{3\mu^2}{2E}\sum_{i,j=1}^{3}\Bigg(\overbrace{-\frac{2}{3}\big(\frac{1}{2}(q_{ij}+q_{ji})\big)\delta_{ij}+(q_{ij}+q_{ji})}^{=A_{ij}}\Bigg)^2\\ &= \frac{3\mu^2}{2E}\sum_{i,j=1}^{3}(A_{ij})^2.\end{aligned}$$

Finally, we are able to compute the deterministic life prediction as (see Figure 3-1):

$$N_{det} = CMB^{-1}\circ RO\circ SD^{-1}\Bigg(\frac{(\sigma_\nu^{el-pl})^2}{E}\Bigg) \tag{3.5}$$

3.2 The Local and Probabilistic Model for LCF

We refer in this section to the probabilistic model for LCF presented in [21] [43], which based on a statistical model for crack initiation.

We consider the mechanical component $\Omega \subset \mathbb{R}^3$ as a bounded, open domain with Lipschitz boundary $\partial\Omega$ and let $\mathcal{C} = (0,\infty] \times \partial\Omega$ be a collection of pairs of times and location.

Definition 3.2.1 ***Radon and counting measures***

Let $\mathcal{B}(\mathcal{C})$ be denotes the Borel-σ-algebra. The space of Radon measures $\mathcal{R}$ is a set of measures γ on the measurable space $(\mathcal{C}, \mathcal{B}(\mathcal{C}))$ such that $\gamma(A) < \infty$ for a bounded $A \in \mathcal{B}(\mathcal{C})$.

A Radon measure $\gamma \in \mathcal{R}$ is a counting measure if $\gamma(B) \in \mathbb{N}_0$, for a bounded and measurable

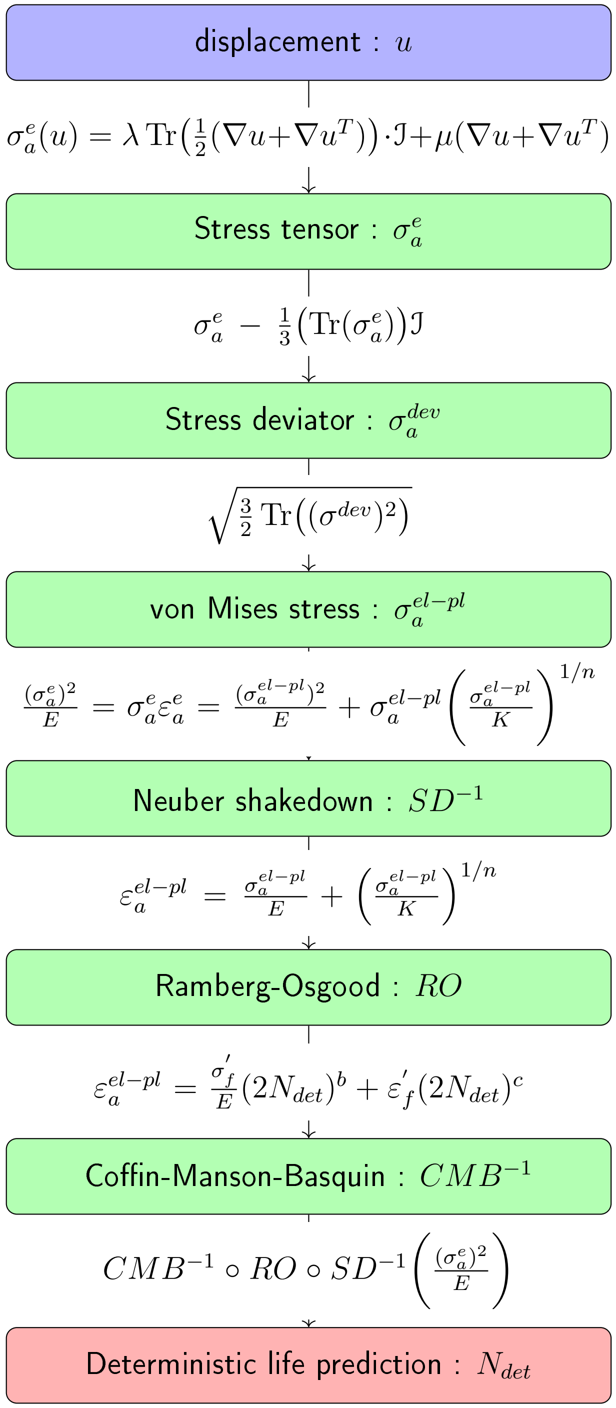


Figure 3-1: Flow diagram of computations of the deterministic life prediction.

$B \subset \mathcal{C}$.

We denote by $\mathcal{R}_c$ the set of all Radon counting measures. Let $C_0(\mathcal{C})$ be the space of continuous functions with compact support on $\mathcal{C}$. The standard σ-algebra on the space $\mathcal{R}_c$ of counting measures generated by the mappings $\gamma \to \int_{\mathcal{C}} f \, d\gamma$ with $f \in C_0(\mathcal{C})$ will be denoted by $\mathcal{R}_c$.

Definition 3.2.2 ***Point process [30]***

A point process defined on $\mathcal{C}$ is a measurable mapping defined on probability space $(X, \mathcal{F}, \mathbb{P})$ taking values in $(\mathcal{R}_c, \mathcal{N}(\mathcal{R}_c))$. This mapping induces a distribution $\mathbb{P}_X$ of X given by

$$\mathbb{P}_X = \mathbb{P}(\{\omega \in \Omega, X(\omega) \in F\}) \text{ for } F \in \mathcal{N}(\mathcal{R}_c).$$

The point process on $\mathcal{C}$ is called a simple point process if its realizations contain no coincident points.
If $\mathbb{P}(\gamma(\{c\}) > 0) = 0$ for all $c \in \mathcal{C}$ holds, we say that the point process is non-atomic.
Let $B_1, .., B_n \in \mathcal{B}(\mathcal{C})$ mutually disjoint. The point process γ has independent increments if the random variables $\gamma(B_1), ..., \gamma(B_n)$ are independent.

The assumption that the point process is non-atomic describe the fact that the probability of failure originating exactly from a location in $\overline{\Omega}$ is equal to zero.

Definition 3.2.3 ([21]) *Let γ be a point process on $\mathcal{C}$.*
We say that γ is a crack initiation process if γ is simple and non-atomic.

The time to crack initiation is defined as

$$\tau(\gamma) = \min\{t > 0 : \gamma(\{(s, x) \in \mathcal{C} : s \leq t\}) > 0\}$$

This last expression is well-defined for the reasons that Radon measures are upper continuous for sets with finite diameter and that the mapping $t \to \gamma(\mathcal{C}_t)$ is right continuous.

Proposition 3.2.4 ***Poisson Point process***

A process γ on $\mathcal{C}$ is Poisson point process if there exist a unique Radon measure $\rho \in \mathcal{R}$ such that

$$\mathbb{P}(\gamma(B) = n) = \exp(-\rho(B))\frac{\rho(B)^n}{n!} \text{ for each } B \in \mathcal{B}(\mathcal{C}) \text{ bounded}$$

where ρ is the intensity measure of γ.

A crack initiation process γ on $\mathcal{C}$ with independent increment is a Poisson point process.

Proof. See [21, 50, 30] . ■

By setting $B = [0, t] \times A$ where $t > 0$ and $A \subset \partial\Omega$, $\rho(B)$ is the expected values of crack initiations in A until time t. Hence

$$\rho(B) = \mathbb{E}(\gamma([0, t] \times A)).$$

Proposition 3.2.5 ***failure distribution***

Let T be the time to crack initiation. The distribution function F_T of T is given by:

$$F_T(t) = 1 - \exp(-\rho(\mathcal{C}_t)), \text{ with } \mathcal{C}_t = \{(x, s) \in \mathcal{C} : s \leq t\}$$

The term $\rho(\mathcal{C}_t)$ is interpreted as the cumulative hazard function.

Proof. See [21] . ■

We would now like to briefly introduce the fundamental principles mentioned in the works [21, 43] that lead to the local, probabilistic model for LCF:

The expected number of crack initiation should depend only on the local and physical quantity i.e. the stress tensor $\sigma(x)$. Furthermore $\rho([0, t] \times A)$ muss be monoton in t because crack will not be disappears as fracture progresses. This leads us to this approach

$$\rho([0,t]\times A)=\int_0^t\int_A \varrho(t,\sigma(x))dAdt,$$

where $\varrho(t,\sigma(x))\geq 0$ describes the local crack initiation intensity of a Poisson point process. Note that we neglect in this approach that ϱ depend on the derivatives of the stress tensor.

Following the proposition 3.2.5, the probability of faiure in $\partial\Omega$ until cycle n is giving by

$$F_N(n)=1-\exp\Big(-\int_0^n\int_{\partial\Omega}\varrho(t,\sigma(x))dAdt\Big),$$

where N represents the cycle of first crack initiation.

The LCF probabilistic model is based on a deterministic CMB life prediction approach described in [44]. By the choice of a Weibull hazard ansatz

$$\varrho(t;\sigma(x))=\frac{\bar{m}}{N_{det}(\sigma(x))}\left(\frac{t}{N_{det}(\sigma(x))}\right)^{\bar{m}-1}, \tag{3.6}$$

the number N of cycles to crack initiation are Weibull distributed. Here $\bar{m}$ is the Weibull shape and the scale field $N_{det}(\sigma),\ x\in\partial\Omega$ is the Weibull scale parameter which is the solution of the CMB equation

$$\varepsilon_a(x)=\frac{\sigma_f^{'}}{E}(2N_{det}(x))^b+\varepsilon_f^{'}(2N_{det}(x))^c,$$

The local and probabilistic model for LCF ist given by the Weibull cumulative distribution function :

$$F_N(t)=1-\exp\Big[-\Big(\frac{t}{\eta}\Big)^{\bar{m}}\Big]$$

where

$$\eta = \left(\int_{\partial\Omega} \frac{1}{N_{det}(\sigma(x))^{\bar{m}}} dA \right)^{-1/\bar{m}}.$$

We are interesting by calculating the sensitivities to the following cost functional for local and probabilistic model for LCF:

$$J_{sur}(\Omega, u) = \int_{\partial\Omega} \frac{1}{N_{det}(\sigma(x))^{\bar{m}}} dA$$

In the next section, we concentrate an the discretization of this surface integral.

3.3 Finite Element Approximations for the LCF-Model

3.3.1 Discretization of the Probabilistic Model

For the numerical approximation of the failure probability we have to compute the cost functional $J_{sur}(\Omega, u)$ which is an integral over the surface $\partial\Omega$. Denote by $\mathcal{N}_h$ a collection of the boundary faces F of finite elements $K = K(F) \in \mathcal{T}_h$ that lie in $\partial\Omega$.

The computation of surface integral $J_{sur}(\Omega, u)$ reduces to evaluating integrals over each element in the collection $\mathcal{N}_h$:

$$\int\limits_{\partial\Omega} \left(\frac{1}{N_{det}(\sigma(x))} \right)^{\bar{m}} dA \approx \sum_{F \in \mathcal{T}_h} \int\limits_{F} \left(\frac{1}{N_{det}(\sigma(x))} \right)^{\bar{m}} dA. \tag{3.7}$$

Let

$$T_F : \widehat{F} \to F$$

be a $\mathcal{C}^1$-diffeomorphism mapping the geometric reference face $\widehat{F} \subset \mathbb{R}^2$ to any face F in $\mathcal{N}_h$ and

$$J_F(\widehat{x}) = \frac{\partial T_F(\widehat{x})}{\partial \widehat{x}} \in \mathbb{R}^{3,2}$$

be the Jacobian matrix of the mapping T_F at $\widehat{x}$ and

$$g_F(\widehat{x}) = (J_F(\widehat{x}))^T J_F(\widehat{x}) \tag{3.8}$$

be the Gram matrix. The change of variables $x = T_K(\widehat{x})$ yields :

$$\int\limits_F \left(\frac{1}{N_{det}(\sigma(x))}\right)^{\bar{m}} dA = \int\limits_{\widehat{F}} \left(\frac{1}{N_{det}(\sigma(T_F(\widehat{x})))}\right)^{\bar{m}} \sqrt{det(g_F(\widehat{x}))} d\widehat{A}. \tag{3.9}$$

We consider a quadrature on $\widehat{F}$ defined by l_q^F Gauß points $\{\widehat{\xi_1^F}, ..., \widehat{\xi_{l_q^F}^F}\}$ and l_q^F weights $\{\widehat{\omega_1^F}, ..., \widehat{\omega_{l_q^F}^F}\}$.

Combining the formulas (3.7) and (3.9) we obtain

$$\begin{aligned} J_{sur}(\Omega, u) &= \sum_{F \in \mathfrak{T}_h} \int\limits_{\widehat{F}} \left(\frac{1}{N_{det}(\sigma(T_F(\widehat{x})))}\right)^{\bar{m}} \sqrt{det(g_F(\widehat{x}))} d\widehat{A} \\ &= \sum_{F \in \mathfrak{T}_h} \sum_{l=1}^{l_q^F} \widehat{\omega_l^F} \left(\frac{1}{N_{det}(\sigma(T_F(\widehat{\xi_l^F})))}\right)^{\bar{m}} \sqrt{det(g_F(\widehat{\xi_l^F}))}. \end{aligned}$$

By setting

$$\omega_{lF} = \widehat{\omega}_l \sqrt{det(g_F(\widehat{\xi_l^F}))} \text{ and } \xi_{lF} = T_F(\widehat{\xi_l^F})$$

we have:

$$J_{sur}^{num}(\Omega, u) \approx \sum_{F \in \mathfrak{T}_h} \sum_{l=1}^{l_q^F} \omega_{lF} \left(\frac{1}{N_{det}(\sigma(\xi_{lF}))}\right)^{-\bar{m}}. \tag{3.10}$$

The value of $J_{sur}(\Omega, u)$ can only be found by computing

$$\sigma(x) = \lambda \nabla \cdot u(x) \mathfrak{I} + 2\mu\varepsilon(u(x)).$$

By expanding u in the global basis functions θ_j, we get

$$u(\xi) = \sum_{j=1}^{N} u_j \theta_j(\xi) = \sum_{K \in \mathfrak{T}_h} \sum_{m=1}^{n_{\mathrm{sh}}} u_{j_{(K,m)}} \widehat{\theta}_m \circ T_K^{-1}(\xi) \tag{3.11}$$

so that

$$\nabla u(\xi) = \sum_{m=1}^{n_{\mathrm{sh}}} u_{\widehat{j}(K,m)} \otimes (J_K(\widehat{\xi})^T)^{-1} \widehat{\nabla} \widehat{\theta}_m(\widehat{\xi})), \quad \text{for } \xi \in K \text{ and } \xi = T_K(\hat{\xi}), \tag{3.12}$$

where

$$J_K(\widehat{\xi}) = \widehat{\nabla} T_K(\hat{\xi}) = \sum_{j=1}^{n_{\mathrm{sh}}} \widehat{\nabla} \widehat{\theta}_j(\widehat{\xi}) X_{K,j}$$

be the Jacobian matrix of the mapping T_K.

From the gradient formula (3.12), we have

$$\nabla \cdot u(\xi) = \sum_{m=1}^{n_{\mathrm{sh}}} \mathrm{tr}\, \Big(u_{\widehat{j}(K,m)} \otimes (J_K(\widehat{\xi})^T)^{-1} \widehat{\nabla} \widehat{\theta}_m(\widehat{\xi})) \Big) \text{ for } \xi \in K \text{ and } \xi = T_K(\hat{\xi}). \tag{3.13}$$

Finally, we can write the probabilistic cost functional in discretized form as

$$J_{sur}(\Omega, u) = \sum_{F \in \mathfrak{T}_h} \sum_{l=1}^{l_q^F} \omega_{lF} \Bigg(CMB^{-1} \Bigg(\varepsilon_a^{el-pl} \Bigg(\sum_{m=1}^{n_{sh}} u_{\widehat{j}(K,m)} \otimes (J_K(\widehat{\xi})^T)^{-1} \widehat{\nabla} \widehat{\theta}_m(\widehat{\xi}) \Bigg) \Bigg) \Bigg)^{\bar{m}}. \tag{3.14}$$

3.3.2 Discretization of the Governing Equation

We return to the discretized governing equation described in (2.6) and we consider the following bilinear form

$$B(u, v) = \int\limits_{\Omega} f \cdot v \, dx + \int\limits_{\partial\Omega_N} g \cdot v \, dA, \forall v \in V_{D,h}^1(\Omega, \mathbb{R}^3)$$

where $u \in V^1_{D,h}(\Omega, \mathbb{R}^3)$.

Similar to the discretization of the probabilistic cost functional, the bilinear form can be discretized in the following way

$$\begin{aligned}
B(u,v) &= \lambda \sum_{K\in\mathfrak{T}_h} \int_K \nabla\cdot u\,\nabla\cdot v dx + 2\mu \sum_{K\in\mathfrak{T}_h}\int_K \varepsilon(u) : \varepsilon(v)\,dx \\
&= \lambda \sum_{K\in\mathfrak{T}_h} \int_K \nabla\cdot u(T_K(\hat{\xi}))\nabla\cdot v(T_K(\hat{\xi}))\det(J_K(\hat{\xi}))\,d\hat{\xi} \\
&+ 2\mu \sum_{K\in\mathfrak{T}_h} \int_K \varepsilon(u(T_K(\hat{\xi}))) : \varepsilon(v(T_K(\hat{\xi})))\det(J_K(\hat{\xi}))\,d\hat{\xi} \\
&\approx \lambda \sum_{K\in\mathfrak{T}_h}\sum_{l=1}^{l_q} \hat{\omega}_l \det(J_K(\hat{\xi}_l))\,\nabla\cdot u(T_K(\hat{\xi}_l))\,\nabla\cdot v(T_K(\hat{\xi}_l)) \\
&+ 2\mu \sum_{K\in\mathfrak{T}_h}\sum_{l=1}^{l_q} \hat{\omega}_l \det(J_K(\hat{\xi}_l))\varepsilon(u(T_K(\hat{\xi}_l))) : \varepsilon(v(T_K(\hat{\xi}_l))),
\end{aligned} \tag{3.15}$$

where $\{\widehat{\xi_1}, ..., \widehat{\xi_{l_q}}\}$ are l_q Gauß points and $\{\widehat{\omega_1}, ..., \widehat{\omega_{l_q}}\}$ are l_q volume quadratures. After setting $\omega_{lK} = \widehat{\omega_l}\det\Big(J_K(\widehat{\xi_l})\Big)$ and $\xi_l^K = T_K(\widehat{\xi_l})$, this can be written as

$$B(u,v) = \underbrace{\lambda \sum_{K\in\mathfrak{T}_h}\sum_{l=1}^{l_q} \omega_{lK}\nabla\cdot u(\xi_l^K)\nabla\cdot v(\xi_l^K)}_{B_1(u,v)} + \underbrace{2\mu \sum_{K\in\mathfrak{T}_h}\sum_{l=1}^{l_q} \omega_{lK}\varepsilon\big(u(\xi_l^K)\big) : \varepsilon\big(v(\xi_l^K)\big)}_{B_2(u,v)} \tag{3.16}$$

For the volume integral we obtain by a similar argument

$$\int_\Omega f\cdot v\,dx = \sum_{K\in\mathfrak{T}_h}\sum_{l=1}^{l_q} \omega_{lK} f(\xi_l^K)\cdot v(\xi_l^K). \tag{3.17}$$

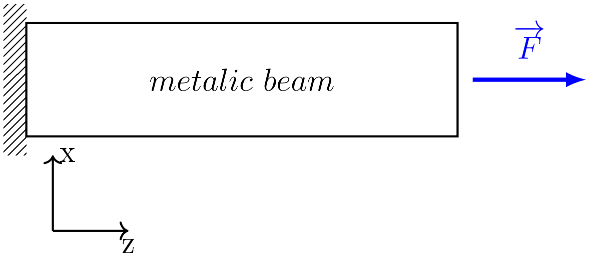


Figure 3-2: A cantilever beam subjected to to an axial force F

The surface force can be discretized as follow

$$\int_{\partial\Omega} g \cdot v \, dA = \sum_{F\in\mathcal{N}_h} \sum_{l=1}^{l_q^F} \omega_{lF} g(\xi_{lF}) \cdot v(\xi_{lF}). \tag{3.18}$$

We will note that the above formulas have to be understood in the sense of approximations if the selected surface and volume quadratures are not exact.

3.4 Numerical Validations and Applications

3.4.1 First Application : Cantilever Beam

Practical problems of linear elasticity are not easy to solve in three dimensional space. This is due on the one hand to the solutions which depend on the geometrical boundary conditions of the problem under consideration. On the other hand, the large number of degrees of freedom of the partial differential equations makes computations slow. In order to validate the code of the numerical computations we need to make a direct comparison of the simulations result with the analytical solutions.

We consider as illustrated in figure (3-2) a short section of a fixed-free cantilever beam of length L and square cross section $w \times w$ subjected to an axial force F stretches by du.

In the following we want to compute analytically the cost functional J_{sur}^{ana} for this case. As mentioned, we apply a normal stress on the positive $z-$face of the metallic beam.

Therefore, the only nonzero stress tensor component is σ_z and we can write:

$$\sigma = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_z \end{bmatrix}.$$

The stress deviator can be computed as

$$\sigma^{dev} = \sigma - \frac{1}{3}\operatorname{tr}(\sigma).\mathcal{I} = \begin{bmatrix} -\frac{1}{3}\sigma_z & 0 & 0 \\ 0 & -\frac{1}{3}\sigma_z & 0 \\ 0 & 0 & \frac{2}{3}\sigma_z \end{bmatrix}$$

Von Mises stress is

$$\sigma^e_\nu = \left(\frac{3}{2}\sum_{i,j=1}^{3} d_{ij}d_{ji}\right)^{1/2},$$

where d_{ij} are the components of the stress deviator tensor σ^{dev}.

We obtain

$$\sigma^e_\nu = \left(\frac{3}{2}\big(d_{11}^2 + d_{22}^2 + d_{33}^2\big)\right)^{1/2} = \left(\frac{3}{2}\big(\frac{1}{9}\sigma_z^2 + \frac{1}{9}\sigma_z^2 + \frac{4}{9}\sigma_z^2\big)\right)^{1/2} = \left(\frac{3}{2}.\frac{6}{9}\sigma_z^2\right)^{1/2} = \sigma_z.$$

The elastic strain is:

$$\epsilon_\nu = RO \circ SD^{-1}\left(\frac{(\sigma^{dev})^2}{E}\right) = RO \circ SD^{-1}\left(\frac{\sigma_z^2}{E}\right).$$

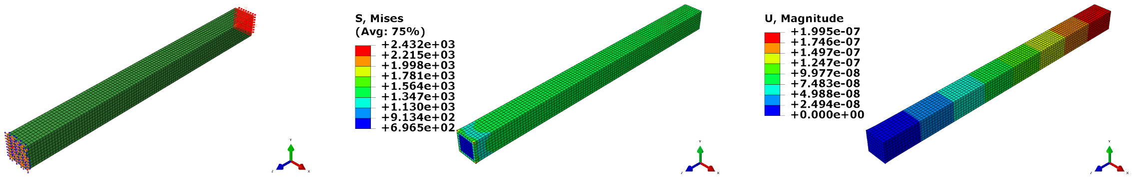


Figure 3-3: (a) Boundary conditions and load by the discretized metallic beam (b) Von Mises stress (c) Displacement

Then, we can write the analytical probabilistic cost functional as:

$$\begin{aligned} J_{sur}^{ana}(\Omega, u) &= \int_{\partial\Omega} [CMB^{-1}(\epsilon_\nu)]^{-m} dA \\ &= (CMB^{-1} \circ RO \circ SD^{-1}\left(\frac{\sigma_z^2}{E}\right))^{-m} \int_{\partial\Omega} dA \\ &= (CMB^{-1} \circ RO \circ SD^{-1}\left(\frac{\sigma_z^2}{E}\right))^{-m}. S(\partial\Omega). \end{aligned} \tag{3.19}$$

where $S(\partial\Omega)$ stands for the surface of Ω.

3.4.2 Numerical Validation : Cantilever Beam

The lenght L of the beam is 10 mm. The surface of the square is $(0.8)^2$ mm^2 and the applied stress is 12 N/mm^2 . The beam will be discretized by applying the finite element method to 6400 brick element of types C3D20R with 20 degrees of freedom, 8 volume quadratures and 4 surface quadratures. The total number of nodes is 30825. Ramberg-Osgood Parameters and deterministic CMB parameters are taken from [10]. The probabilistic CMB parameters needed here are obtained from the deterministic CMB parameters by using the following scaling relation

$$\frac{\sigma'_f(|\partial\Omega_1|)}{\sigma'_f(|\partial\Omega_2|)} = \left(\frac{|\partial\Omega_1|}{|\partial\Omega_2|}\right)^{\frac{b}{m}}, \quad \frac{\varepsilon'_f(|\partial\Omega_1|)}{\varepsilon'_f(|\partial\Omega_2|)} = \left(\frac{|\partial\Omega_1|}{|\partial\Omega_2|}\right)^{\frac{c}{m}} \tag{3.20}$$

described in [46, 43].

For solving the FEA model we use the commercial solver ABAQUS 6.1 on a laptop with Intel Core i7-3632QM CPU @ 2.20 GHz with 12GB RAM. J_{sur}^{num} is computed over all 3328 faces (rectangles) on the beam boundaries. A self implemented FEM-tool in R language is used for this computation. The Weibull shape parameter is $\bar{m} = 2$.

As comparison between the analytical probabilistic cost functional computed with the formula (3.10) and the numerical probabilistic cost functional computed in (3.19) we have:

$$\frac{J_{sur}^{ana}}{J_{sur}^{num}} = \frac{2.928616e - 16}{2.928615e - 16} \simeq 1.$$

This type of validation using analytical method allow us to check the implemented code used for the numerical computation.

3.4.3 Second Application : Turbo Compressor

As a second application, we study the crack initiation for a radial turbo compressor of a jet engine. This model is a part of CalculiX package, which is developed by employees of MTU Aero Engines in Munich. The compressor is loaded by pure centrifugal force and is constructed of casted aluminum alloy AlMgSi6082, containing roughly 97 $wt\%$ Al, roughly 1 $wt\%$ Si and Mg along with minor contributions of Mn, Fe and Cr. We refer to [10] for more chemical composition details.

We neglect in this computation air pressure and we consider the volume forces caused by the centrifugal load with a rotation speed of 110 000 rpm and a specific density of the material of 2.65 g/cm^3. The material parameters used in this model are Young's modulus $E = 70000\ MPa$ and Poisson's ratio $\nu = 0.3$. The Lamè coefficient are $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} = 40385\ MPa$ and $\mu = \frac{E}{2(1+\nu)} = 26923\ MPa$.

The discretized cost functional J_{sur} is computed for a model jet engine radial turbo

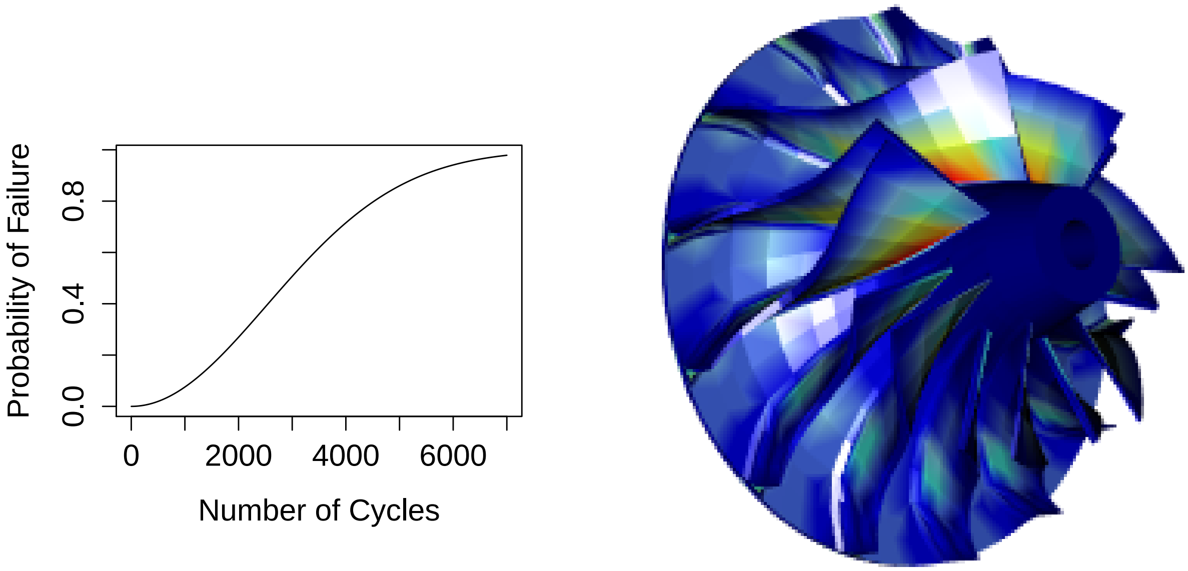


Figure 3-4: Probability of failure over the number of load cycles (right) for a radial compressor model. (a) PoF over the number of load cycles for the compressor model. (b) crack initiation intensity in logarithmic color code from blue (low) to red (high) .

compressor obtained from [52]. The preprocessing data (coordinates, elements connectivity and displacements) are handled using the free FEA-Software CalculiX. The compressor is discretized using a 20 node hexahedral element with reduced quadratures ($he20r$) into 9464 elements. J_{sur} is calculated over all 6356 faces (rectangles) on the compressor boundary.

The material parameters have been taken from [10]. Ramberg-Osgood Parameters are $n' = 0.064$ and $K = 443.9$. Table 3.1 shows CMB parameters and we apply the scaling relation (3.20) to pass from deterministic to probabilistic CMB parameters. The Weibull shape parameter is $\bar{m} = 2$, which is a usual value for polycristalline metal. We followed the approach described in (3.20) to adapt the CMB-parameters to our compressor model because the size of specimens have a non-negligible influence on crack-initiation life.

As described in the last section 3.4.1, the same solver and machine is used to solve the

	σ'_f(MPa)	ϵ'_f	b	c	$\bar{m}$	$\lvert\partial\Omega\rvert$ (mm^2)
Deterministic	487	0.209	-0.07	-0.593	–	377
Probabilistic	599	1.213	-0.07	-0.593	2	1

Table 3.1: Material parameters for the probabilistic model and their deterministic counterparts (taken from [10]). The parameter $\bar{m}$ is a guess on the basis of probabilistic investigation of other polycristalline metals.

FEA model for u. The time needed for this computations is 28 second. The maximum von Mises stress at the bore is about 310 MPa. Many scripts implemented in R.3.2.1 are used to compute the surface quadratures for the calculation of $J(\Omega, u)$ and take on a single core a 19 second execution time. We will take into account that the compressor is composed from seven rotated sectors. Hence the obtained J-value will be multiplied by 7 and the resulting total value is 7.8541×10^{-8}. The scale Weibull parameter is $\eta = (7.8541 \times 10^{-8})^{-1/2} = 3568$ cycles and corresponds to the $1 - \frac{1}{e} \approx 63\%$ quantiles of crack initiation life. Figure 3-4 shows the probability of failure (PoF) over the number of cycles t and (b) visualizes the local failure intensity described in (3.6).

Chapter 4

Shape Sensitivity Analysis

In this chapter we present the adjoint sensitivity method that is used to study the effects of varied design parameters to the variation of the objective functional J_{sur}. The following section begins with an introduction to the computation of the total sensitivity, where both methods - the direct and the adjoint method - are derived. Then, we describe the details of the calculation and implementation of the partial derivative terms in the total sensitivity equation. Finally, we present results of the application of this sensitivity computation method to finite element models.

4.1 Sensitivity Analysis

4.1.1 Shape Sensitivity : Direct Method

The goal of applying the sensitivity analysis is to investigate how a functional of interest varies with respect of some perturbations on the shape geometry. The probabilistic functional $J_{sur}(\Omega, u)$ developed in the last chapter not only depends on the design variables, but also on the displacement field. After the discretization of the shape with the FEM, let the set $X = \{X_1, \ldots, X_N\}$ indicate the mesh nodes (design variables). Let

$U = (u_j)_{j=1,\ldots,N}$, $u_j \in \mathbb{R}$ represent the global displacement with

$$U = \sum_{j=1}^{N} u_j \theta_j,$$

where θ_j are the global shape functions. Instead of $J_{sur}(\Omega, u)$ we can equivalently write $J(X, U)$. By rewriting the governing equation described in (2.6) we have:

$$B(X)U = F(X)$$

where

$$B(X)_{(r,j),(s,k)} = B(e_r\theta_j, e_s\theta_k),$$

is a $3 \times N \times 3 \times N$ tensor representing the stiffness matrix and

$$F_{(r,j)} = \int\limits_{\Omega} f \cdot e_r\theta_j \, dx + \int\limits_{\partial\Omega_N} g \cdot e_r\theta_j \, dA,$$

is a $3 \times N$ tensor representing the load vector. The vectors e_s for $s = 1, 2, 3$ are the canonial basis on $\mathbb{R}^3$. By the following expression

$$R(X, U(X)) = B(X)U - F(X) = 0$$

we denote the residual of the this governing equation. Remark that the displacement U is a $3 \times N$ tensor and depends on the geometry variable X which is a $3 \times N$ tensor.

The total derivative of $R(X, U(X))$ with respect to X is

$$\frac{dR(X, U(X))}{dX} = \frac{\partial R(X, U(X))}{\partial X} + \frac{\partial R(X, U(X))}{\partial U}\frac{\partial U(X)}{\partial X} = 0. \tag{4.1}$$

The term $\frac{\partial R(X,U(X))}{\partial U}\frac{\partial U(X)}{\partial X}$ in the last equation can be seen as a contraction of tensor slots generated by the partial U-differentiation in $\frac{\partial R(X,U(X))}{\partial U}$ and the U-slots in $\frac{\partial U(X)}{\partial X}$. Applying the chain rule, the total sensitivity is:

$$\frac{dJ(X,U(X))}{dX} = \frac{\partial J(X,U(X))}{\partial X} + \frac{\partial J(X,U(X))}{\partial U}\frac{\partial U(X)}{\partial X}. \tag{4.2}$$

By rewriting the equation (4.1) as

$$\frac{\partial R(X,U(X))}{\partial U}\frac{\partial U(X)}{\partial X} = -\frac{\partial R(X,U(X))}{\partial X}, \tag{4.3}$$

we can solve for $\frac{\partial U(X)}{\partial X}$. By substituting this result into the total sensitivity equation (4.2), we obtain

$$\frac{dJ(X,U(X))}{dX} = \frac{\partial J(X,U(X))}{\partial X} - \frac{\partial J(X,U(X))}{\partial U}\left[\frac{\partial R(X,U(X))}{\partial U}\right]^{-1}\frac{\partial R(X,U(X))}{\partial X}. \tag{4.4}$$

The approach, where the equation (4.3) is solved and the result is used in the expression of the equation (4.4), is called the *direct* method.

4.1.2 Shape Sensitivity : Adjoint Method

The adjoint method used in our work is based on a discretized formulation that derived from the governing partial differential equation. This is opposed to the continuous approach where the adjoint equation is first derived then discretized [47]. The Lagrangian of the discretized problem is defined as

$$\mathscr{L}(X,U,\Lambda) = J(X,U) - \Lambda^T\left(B(X)U - F(X)\right), \tag{4.5}$$

where in this context the adjoint state Λ is a Lagrange multiplier. Remark that $\Lambda = (\lambda_j)_{j\in\{1,\dots,N\}}$ is a $N \times 3$ tensor and if Dirichlet boundary conditions are satisfied for $U =$

(u_j), then we have the same for $\Lambda = (\lambda_j)$ i.e.

$$If\ u_j = 0 \text{ for } X_j \in \partial\Omega_D \Rightarrow \lambda_j = 0.$$

After differentiation with respect to the displacement tensor we have

$$\frac{\partial \mathscr{L}(X, U, \Lambda)}{\partial U} = \frac{\partial J(X, U)}{\partial U} - \Lambda^T B(X) = 0. \tag{4.6}$$

Since the stiffness matrix (tensor) is symmetric in the (j, r) and (k, l) indices, we can write the *adjoint* equation as:

$$B(X)\Lambda = \left(\frac{\partial J(X, U)}{\partial U}\right)^T. \tag{4.7}$$

In the previous equation, the components of the right-hand side are usually non-zero only for those element-points on the surface mesh. Remark that the adjoint equation has the same stiffness matrix as the governing equation. The right-hand side in the adjoint equation can be seen as a pseudo load. The total derivative of $J(X, U(X))$ is

$$\frac{dJ(X, U(X))}{dX} = \frac{\partial \mathscr{L}(X, U, \Lambda)}{\partial X} = \frac{\partial J(X, U)}{\partial X} - \Lambda^T \left(\frac{\partial B(X)}{\partial X} U - \frac{\partial F(X)}{\partial X}\right). \tag{4.8}$$

4.1.3 Partial Derivatives Calculation: Direct Approach

In this section, we describe the details of calculation of the partial derivatives needed to solve the total derivative by using the adjoint method. We start by calculating the partial derivative of the objective functional J with respect to the displacement U.

Calculating $\frac{\partial J(X,U)}{\partial U}$:

Let

$$\widehat{j} : \mathcal{T}_h \times \{1, \ldots, n_{\text{sh}}\} \to \{1, \ldots, N\} \tag{4.9}$$

be the connectivity mapping, which gives the relationship between the finite element $K \in \mathcal{T}_h$ with its local degree of freedom in $\{1, \ldots, n_{sh}\}$ and the index in $\{1, \ldots, N\}$ defining the global degree of freedom. A restriction of the mapping $\widehat{\jmath}$ to the set $\{(K, 1), \ldots, (K, n_k)\}$ is denoted by

$$\widehat{\jmath}_K : \{1, \ldots, n_{\text{sh}}\} \to \{1, \ldots, N\}$$

where the set $\{1, \ldots, n_{sh}\}$ is identified with $\{(K, 1), \ldots, (K, n_k)\}$. Reciprocally, for a given global index $j \in \{1, 2, \ldots, N\}$ we have $\widehat{\jmath}^{-1}(j) = \{(K_1, m_1), ..., (K_f, m_f)\}$ where $K_1, ..., K_f \in \mathcal{T}_h$ and $m_1, ..., m_f \in \{1, 2, ..., n_{\text{sh}}\}$. We note with $\widehat{\jmath}^{-1}(j)_1 = \{K_1, \ldots, K_f\}$ the set projection to the first component.

For $k = 1, 2, 3$ let u_{jk} denote the x, y, z coordinate of the global degree of freedom u_j. The partial derivative of J with respect to the displacement is

$$\begin{aligned} \frac{\partial J(X,U)}{\partial u_{jk}} = & \sum_{\substack{F \in \mathcal{N}_h \\ j \in \widehat{\jmath}_{K(F)}(\{1,\ldots,n_{\text{sh}}\})}} \sum_{l=1}^{l_q^F} \omega_{lF} \\ & \times \frac{\partial}{\partial u_{jk}} \left[\text{CMB}^{-1} \left(\varepsilon_a^{\text{el-pl}} \left(\sum_{m=1}^{n_{\text{sh}}} u_{\widehat{\jmath}(K(F),m)} \otimes ((J_{K(F)}^T(\widehat{\xi}_l^F))^{-1} \widehat{\nabla} \widehat{\theta}_m(\widehat{\xi}_l^F)) \right) \right) \right]^{-\bar{m}}. \end{aligned} \tag{4.10}$$

We use the following notation to wrap the last formula:

$$q = q_{lF} = q_{lF}(U, X) = \nabla u(\xi_l^F) = \sum_{m=1}^{n_{\text{sh}}} u_{\widehat{\jmath}_{(K(F),m)}} \otimes (J_{K(F)}^T(\widehat{\xi}_l^F))^{-1} \widehat{\nabla} \widehat{\theta}_m(\widehat{\xi}_l)), \; l = 1, \ldots, l_q^F, \tag{4.11}$$

with $\xi_l^F = T_{K_F}(\hat{\xi}_l)$.

Hence, we have

$$\frac{\partial}{\partial u_{jk}}[\mathrm{CMB}^{-1}(\varepsilon_a^{\mathrm{el-pl}}(q_{lF}))]^{-\bar{m}} = -\bar{m}[\mathrm{CMB}^{-1}(\varepsilon_a^{\mathrm{el-pl}}(q_{lF}))]^{-\bar{m}-1}\frac{\partial}{\partial u_{jk}}[\mathrm{CMB}^{-1}(\varepsilon_a^{\mathrm{el-pl}}(q_{lF}))]. \tag{4.12}$$

To continue, we need to compute the last term in (4.12)

$$\frac{\partial}{\partial u_{jk}}[\mathrm{CMB}^{-1}(\varepsilon_a^{\mathrm{el-pl}}(q_{lF}))] = \frac{\frac{\partial}{\partial u_{jk}}\big(\varepsilon_a^{\mathrm{el-pl}}(q_{lF})\big)}{\frac{\partial \mathrm{CMB}}{\partial \varepsilon_a^{\mathrm{el-pl}}}(\mathrm{CMB}^{-1}(\varepsilon_a^{\mathrm{el-pl}}(q_{lF})))}\ , \tag{4.13}$$

A calculation of the numerator term yields

$$\frac{\partial}{\partial u_{jk}}\big(\varepsilon_a^{\mathrm{el-pl}}(q_{lF})\big) = \frac{\partial \varepsilon_a^{\mathrm{el-pl}}(q_{lF})}{\partial q_{lF}} : \frac{\partial q_{lF}}{\partial u_{jk}} = \mathrm{tr}\left(\Big(\frac{\partial \varepsilon_a^{\mathrm{el-pl}}(q_{lF})}{\partial q_{lF}}\Big)^T \frac{\partial q_{lF}}{\partial u_{jk}}\right), \tag{4.14}$$

where for $s, n = 1, 2, 3$,

$$\begin{aligned}
\frac{\partial (q_{lF})_{sn}}{\partial u_{kj}} &= \frac{\partial}{\partial u_{kj}}\left(\sum_{m=1}^{n_{\mathrm{sh}}} u_{\widehat{j}(K(F),m)s}\left((J_{K(F)}^T(\widehat{\xi}_l^F))^{-1}\widehat{\nabla}\widehat{\theta}_m(\widehat{\xi}_l^F)\right)_n\right) \\
&= \sum_{m=1}^{n_{\mathrm{sh}}} \frac{\partial u_{\widehat{j}(K(F),m)s}}{\partial u_{jk}}\left((J_{K(F)}^T(\widehat{\xi}_l^F))^{-1}\widehat{\nabla}\widehat{\theta}_m(\widehat{\xi}_l^F)\right)_n \\
&= \begin{cases} \delta_{sk}\left((J_{K(F)}^T(\widehat{\xi}_l^F))^{-1}\widehat{\nabla}\widehat{\theta}_{\widehat{j}^{-1}_{K(F)}(j)}(\widehat{\xi}_l^F)\right)_n & \text{if } j \in \widehat{j}_{K(F)}(\{1,\dots,n_{\mathrm{sh}}\}) \\ 0 & \text{otherwise} \end{cases},
\end{aligned} \tag{4.15}$$

where δ_{sk} denotes Kronecker's delta.

We compute the partial derivative of $\varepsilon_a^{\text{el-pl}}$ with respect to q_{sm}:

$$\frac{\partial \varepsilon_a^{\text{el-pl}}}{\partial q_{sm}} = \frac{d\Bigg(\mathrm{RO} \circ \bar{\mathrm{SD}}^{-1}\left(\frac{(\sigma_a)^2}{E}\right)\Bigg)}{d\left(\frac{(\sigma_a)^2}{E}\right)} \cdot \frac{\partial}{\partial q_{sm}}\bigg(\frac{(\sigma_a)^2}{E}\bigg). \tag{4.16}$$

The derivative of the composition $\mathrm{RO} \circ \bar{\mathrm{SD}}^{-1}(\cdot)$ of the functions $\mathrm{RO}(\cdot)$ and $\bar{\mathrm{SD}}^{-1}(\cdot)$ is easily calculated. We continue the calculation

$$\frac{\partial}{\partial q_{sm}}\bigg(\frac{(\sigma_a)^2}{E}\bigg) = \frac{3\mu^2}{4E}\sum_{i,j=1}^{3} A_{ij}\frac{\partial A_{ij}}{\partial q_{sm}} \tag{4.17}$$

where $A_{ij} = \Bigg(-\frac{2}{3}\big(\frac{1}{2}(q_{ij}+q_{ji})\big)\delta_{ij} + q_{ij} + q_{ji}\Bigg)$.

We thus obtain

$$\frac{\partial A_{ij}}{\partial q_{s,m}} = -\frac{2}{3}\delta_{ij}\delta_{sm} + \delta_{is}\delta_{jm} + \delta_{js}\delta_{im}. \tag{4.18}$$

The left hand side of (4.17) can be written as

$$\frac{3\mu^2}{4E}\sum_{i,j=1}^{3}\Bigg(-\frac{2}{3}\big(\frac{1}{2}(q_{ij}+q_{ji})\big)\delta_{ij} + q_{ij} + q_{ji}\Bigg)\big(-\frac{2}{3}\delta_{ij}\delta_{sm} + \delta_{is}\delta_{jm} + \delta_{js}\delta_{im}\big). \tag{4.19}$$

In the next step we describe the algorithm used to compute the partial derivative $\frac{\partial J(X,U)}{\partial U}$. Note that here it suffices to calculate all partial derivatives for finite elements $K \in \mathcal{T}_h$ with at least one surface face $F \in \mathcal{N}_h$, since all other partial derivatives of $J(X,U)$ give the value zero.

```
Data: FE global node set $X$, FE connectivity and surface element tables,
FE shape functions $\hat{\theta}_j$ and gradients $\nabla\hat{\theta}_j$ and FE solution $U = (u_j)$,
FE surface quadrature points and weights, elasticity and lifing material constants.
Result: A $(3 \times n_{\text{sh}} \times N_{el}^{\partial})$ tensor containing $\frac{\partial J}{\partial U_{\text{loc}}}$
initialization $\frac{\partial J}{\partial U_{\text{loc}}} \leftarrow 0$;
for all faces $F \in \mathcal{N}_h$ do
    initialize a $3 \times n_{\text{sh}} \times l_q$-tensor $\frac{\partial J}{\partial U}^F \leftarrow 0$;
    for all local degrees of freedom, $k = 1, \ldots, n_{\text{sh}}$ do
        for all surface quadrature points $\hat{\xi}_l^F$, $l = 1, \ldots, l_q^{\partial}$ do
            Calculate the derivative of (3.12) and (3.13) with respect to $u_k$ at the quadrature point $\hat{\xi}_l^F$;
            Calculate the derivative of the stress tensor $\sigma(T_{K(F)}(\hat{\xi}_l^F))$ with respect to $u_k$;
            Calculate the derivative of $\sigma_v$ with respect to $u_k$;
            Use (3.1) as well as (3.2), (3.3) and (3.5) to calculate the $u_k$-derivative of $\text{Ni}_{\text{det}}(\sigma(T_{K(F)}(\hat{\xi}_l^F)))$ ;
            Use this to calculate the derivative of $\left(\frac{1}{Ni_{\text{det}}(\sigma(T_{K(F)}(\hat{\xi}_l^F)))}\right)^{\bar{m}}$;
            Multiply this with the surface quadrature weight $\hat{\omega}_l^F$ and with the Gram determinant $\sqrt{g_F(\hat{\xi}_l^F)}$;
            Store the result in $\frac{\partial J}{\partial U}^F[\cdot, k, l]$;
        end
    end
    Sum up the $(3 \times n_{\text{sh}} \times l_q^{\partial})$ tensor $\frac{\partial J}{\partial U}^F$ obtained over the quadratures;
    Augment $\frac{\partial J}{\partial U_{\text{loc}}}[\cdot, \cdot, K(F)]$ by the result;
end
```

Algorithm 1: Compute $\frac{\partial J}{\partial U_{\text{loc}}}$

The algorithm 1 provides a surface face-wise separate computation of the partial derivative of the cost functional J with respect the the local vector of displacements. The local derivatives $\frac{\partial J}{\partial U_{\text{loc}}}$ are stored in an array of dimension $(3 \times n_{\text{sh}} \times N_{el}^{\partial})$.
The next step is to assemble those local derivatives based on the connectivity list to obtain the global partial derivative. This assembling process will be described in the following algorithm.

```
Data: A $3 \times n_{\text{sh}} \times N_{el}^{\partial}$ tensor,
FE connectivity table and
surface element table
Result: A $3 \times N$ matrix containing $\frac{\partial J}{\partial U}$

initialization $\frac{\partial J}{\partial U} \leftarrow 0$;
for all elements $K \in \mathcal{T}_h^{\partial}$ do
    initialize a $3 \times n_{\text{sh}} \times l_q$-tensor ${\frac{\partial J}{\partial X}}^F \leftarrow 0$;
    for all local degrees of freedom $k = 1, \ldots, n_{\text{sh}}$ do
        $\frac{\partial J}{\partial U}[\cdot, \widehat{j}(K,m)] \leftarrow \frac{\partial J}{\partial U}[\cdot, \widehat{j}(K,m)] + \frac{\partial J}{\partial U_{\text{loc}}}[\cdot, k, K]$;
    end
end
```

Algorithm 2: Assembling $\frac{\partial J}{\partial U_{\text{loc}}}$ to $\frac{\partial J}{\partial U}$

Calculating $\frac{\partial J(X,U)}{\partial X}$:

We use the same abbreviation as in the last computation of $\frac{\partial J(X,U)}{\partial U}$. The partial derivative of $J(X,U)$ with respect to the global jth geometry mesh node X_{ji}, where $i = 1, 2, 3$ and

$j = 1, 2, \ldots, N$ is:

$$\begin{aligned}\frac{\partial J(X,U)}{\partial X_{ji}} &= \sum_{\substack{F \in \mathcal{N}_h \\ j \in \widehat{j}_{K(F)}(\{1,\ldots,n_{\mathrm{sh}}\})}} \sum_{l=1}^{l_q^F} \frac{\partial}{\partial X_{ji}} \left(\omega_{lF} \left(\mathrm{CMB}^{-1} \left(\varepsilon_a^{\mathrm{el-pl}}(q_{lF}) \right) \right)^{-\bar{m}} \right) \\ &= \sum_{\substack{F \in \mathcal{N}_h \\ j \in \widehat{j}_{K(F)}(\{1,\ldots,n_{\mathrm{sh}}\})}} \sum_{l=1}^{l_q^F} \Bigg[\frac{\partial \omega_{lF}}{\partial X_{ji}} \Big(\mathrm{CMB}^{-1}(\varepsilon_a^{\mathrm{el-pl}}(q_{lF})) \Big)^{-\bar{m}} \\ &+ \omega_{lF} \frac{\partial}{\partial X_{ji}} \Big(\Big(\mathrm{CMB}^{-1}(\varepsilon_a^{\mathrm{el-pl}}(q_{lF})) \Big)^{-\bar{m}} \Big) \Bigg].\end{aligned} \tag{4.20}$$

We start with computing the partial derivative of $\frac{\partial \omega_{lF}}{\partial X_{ji}}$

$$\begin{aligned}\frac{\partial \omega_{lF}}{\partial X_{ji}} &= \frac{\partial}{\partial X_{ji}} \left(\widehat{\omega}_l \sqrt{\det(g_F(\widehat{\xi}_l^F))} \right) \\ &= \widehat{\omega}_l \frac{1}{2} \big(\det(g_F(\widehat{\xi}_l^F)) \big)^{-1/2} \frac{\partial}{\partial X_{ji}} (\det(g_F(\widehat{\xi}_l^F))).\end{aligned} \tag{4.21}$$

Then, we compute the derivative of the determinant

$$\frac{\partial}{\partial X_{ji}} \Big(\det \Big(g_F(\widehat{\xi}_l^F) \Big) \Big) = \det \Big(g_F(\widehat{\xi}_l^F) \Big) \operatorname{tr} \left(\Big(g_F(\widehat{\xi}_l^F) \Big)^{-1} \frac{\partial g_F(\widehat{\xi}_l^F)}{\partial X_{ji}} \right), \tag{4.22}$$

where $g_F(\widehat{\xi}_l^F) = J_F(\widehat{\xi}_l^F)^T J_F(\widehat{\xi}_l^F)$ is the Gram matrix and

$$\begin{aligned}J_F(\widehat{\xi}_l^F) &= \frac{\partial T_K(\widehat{\xi}_l^F)}{\partial \widehat{X}^F} \\ &= \sum_{r=1}^{n_{\mathrm{sh}}} \begin{pmatrix} X_{r1}^{K(F)} \frac{\partial \widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X}_1^F} & X_{1r}^{K(F)} \frac{\partial \widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X}_2^F} \\ X_{r2}^{K(F)} \frac{\partial \widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X}_1^F} & X_{r2}^{K(F)} \frac{\partial \theta_r(\widehat{\xi}_l^F)}{\partial \widehat{X}_2^F} \\ X_{r3}^{K(F)} \frac{\partial \widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X}_1^F} & X_{r3}^{K(F)} \frac{\partial \widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X}_2^F} \end{pmatrix}.\end{aligned} \tag{4.23}$$

Note that $\widehat{X}_i^F$, $i = 1, 2$, stand for the coordinates on the two dimensional reference face

$\widehat{F}$ corresponding to F in $\widehat{K}$.

The derivative of the Gram matrix with respect to X_{ji} is thus given by

$$\frac{\partial g_F(\widehat{\xi}_l^F)}{\partial X_{ji}} = \frac{\partial}{\partial X_{ji}}\big(J_F(\widehat{\xi}_l^F)\big)^T J_F(\widehat{\xi}_l^F) + \big(J_F(\widehat{\xi}_l^F)\big)^T \frac{\partial}{\partial X_{ji}}\big(J_F(\widehat{\xi}_l^F)\big). \tag{4.24}$$

Furthermore, for $s = 1, 2, 3$ and $k = 1, 2$ we have

$$\begin{aligned}\frac{\partial}{\partial X_{ji}}\big(J_F(\widehat{\xi}_l^F)_{sk}\big) &= \sum_{r=1}^{n_{\mathrm{sh}}} \frac{\partial}{\partial X_{ji}} \left(X_{rs}^{K(F)} \frac{\partial \theta_r(\widehat{\xi}_l^F)}{\partial \widehat{X_k}} \right) \\ &= \begin{cases} \delta_{is} \frac{\partial \widehat{\theta}_{\widehat{j}^{-1}_{K(F)}(j)}(\widehat{\xi}_l^F)}{\partial \widehat{X_k}} & \text{if } j \in \widehat{j}_{K(F)}(\{1, \dots, n_{\mathrm{sh}}\}) \\ 0 & \text{otherwise} \end{cases}.\end{aligned} \tag{4.25}$$

Thus the computation of the first term on the right hand side of (4.20) is finished.

For the computation of the second term in (4.20), we take the partial derivative

$$\frac{\partial}{\partial X_{ji}}\Big[\mathrm{CMB}^{-1}(\varepsilon_a^{\mathrm{el-pl}}(q_{lF}))\Big]^{-\bar{m}} = \frac{-\bar{m}[\mathrm{CMB}^{-1}(\varepsilon_a^{\mathrm{el-pl}}(q_{lF}))]^{-\bar{m}-1}}{\frac{\partial \mathrm{CMB}}{\partial \varepsilon_a^{\mathrm{el-pl}}}(\mathrm{CMB}^{-1}(\varepsilon_a^{\mathrm{el-pl}}(q_{lF})))} \frac{\partial}{\partial X_{ji}}\big(\varepsilon_a^{\mathrm{el-pl}}(q_{lF})\big), \tag{4.26}$$

with

$$\frac{\partial}{\partial X_{ji}}\big(\varepsilon_a^{\mathrm{el-pl}}(q_{lF})\big) = \frac{\partial \varepsilon_a^{\mathrm{el-pl}}(q_{lF})}{\partial q_{lF}} : \frac{\partial q_{lF}}{\partial X_{ji}}. \tag{4.27}$$

We will note that the symbol : in the above equation stands for the contraction of both q indices. Next we have to compute

$$\begin{aligned}\frac{\partial q_{lF}}{\partial X_{ji}}(X) &= \frac{\partial}{\partial X_{ji}}\left(\sum_{m=1}^{n_{\mathrm{sh}}} u_{\widehat{j}(K(F),m)} \otimes ((J_{K(F)}^T(\widehat{\xi}_l^F))^{-1} \widehat{\nabla}\widehat{\theta}_m(\widehat{\xi}_l^F))\right) \\ &= \sum_{m=1}^{n_{\mathrm{sh}}} u_{j(K(F),m)} \otimes \Big(\frac{\partial}{\partial X_{ji}}(J_{K(F)}^T(\widehat{\xi}_l^F))^{-1}\Big)\widehat{\nabla}\widehat{\theta}_m(\widehat{\xi}_l^F),\end{aligned} \tag{4.28}$$

where

$$\frac{\partial}{\partial X_{ji}}\big(J_K^T(\widehat{\xi}_l^F)\big)^{-1} = -\big(J_K^T(\widehat{\xi}_l^F)\big)^{-1}\frac{\partial\big(J_K^T(\widehat{\xi}_l^F)\big)}{\partial X_{ji}}\big(J_K^T(\widehat{\xi}_l^F)\big)^{-1}. \tag{4.29}$$

The Jacobian matrix has the form

$$J_K(\widehat{\xi}_l^F) = \begin{pmatrix} \frac{\partial X_1^K}{\partial \widehat{X_1}} & \frac{\partial X_1^K}{\partial \widehat{X_2}} & \frac{\partial X_1^K}{\partial \widehat{X_3}} \\ \frac{\partial X_2^K}{\partial \widehat{X_1}} & \frac{\partial X_2^K}{\partial \widehat{X_2}} & \frac{\partial X_2^K}{\partial \widehat{X_3}} \\ \frac{\partial X_3^K}{\partial \widehat{X_1}} & \frac{\partial X_3^K}{\partial \widehat{X_2}} & \frac{\partial X_3^K}{\partial \widehat{X_3}} \end{pmatrix}$$

$$= \sum_{r=1}^{n_{\mathrm{sh}}} \begin{pmatrix} X_{r1}^K \frac{\partial\widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X_1}} & X_{r1}^K \frac{\partial\widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X_2}} & X_{r1}^K \frac{\partial\widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X_3}} \\ X_{r2}^K \frac{\partial\widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X_1}} & X_{r2}^K \frac{\partial\widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X_2}} & X_{r2}^K \frac{\partial\widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X_3}} \\ X_{r3}^K \frac{\partial\widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X_1}} & X_{r3}^K \frac{\partial\widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X_2}} & X_{r3}^K \frac{\partial\widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X_3}} \end{pmatrix}. \tag{4.30}$$

Finally, we get

$$\begin{aligned} \frac{\partial\big(J_K^T(\widehat{\xi}_l^F)_{sk}\big)}{\partial X_{ji}} &= \sum_{r=1}^{n_{\mathrm{sh}}} \frac{\partial X_{rs}^K}{\partial X_{ji}} \frac{\partial\widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X_k}} \\ &= \begin{cases} \delta_{is}\frac{\partial\widehat{\theta}_{\widehat{j}_K^{-1}(j)}(\widehat{\xi}_l^F)}{\partial \widehat{X_k}}, & \text{if } j \in \widehat{j}_K(\{1,\dots,n_{\mathrm{sh}}\}) \\ 0 & \text{otherwise} \end{cases}. \end{aligned} \tag{4.31}$$

This finishes the computation of the second term.

We follow the same approach as in the algorithm 1 to calculate the partial derivative of J with respect to X. Note that Gram determinant $g_F(\hat{\xi}_l^F)$ and the Jacobi determinant $J_{K(F)}(\hat{\xi}_l)$ depend on the mesh Node X_j with $j = 1,\dots,N$.

```
Data: Same as in Algorithm 1
Result: A $(3 \times n_{\text{sh}} \times N_{el})$ tensor containing $\frac{\partial J}{\partial X_{\text{loc}}}$
initialization $\frac{\partial J}{\partial X_{\text{loc}}} \leftarrow 0$;
for all faces $F \in \mathcal{N}_h$ do
    initialize a $3 \times n_{\text{sh}} \times l_q$-tensor ${\frac{\partial J}{\partial X}}^F \leftarrow 0$;
    for all local degrees of freedom, $k = 1, \dots, n_{\text{sh}}$ do
        for all surface quadrature points $\hat{\xi}_l^K$, $l = 1, \dots, l_q$ do
            Calculate the derivative of Jacobian matrix with respect to $X_k^{K(F)}$;
            Calculate the derivative of (3.12) and (3.13) with respect to $X_k^{K(F)}$
             using (3.11);
            Calculate the $X_k$ derivative of $\sqrt{\det g_F(\hat{\xi}_l^F)}$ using (3.8) and (3.11);
            Multiply the result with $\left(\frac{1}{\text{Ni}_{\text{det}}}\right)^{\bar{m}}$ and $\widehat{\omega}_{lF}$;
            Store the result in ${\frac{\partial J}{\partial X}}^F[\cdot, k, l]$;
            Follow the steps 2–6 of Algorithm 1 analogously with $u_k$ replaced by
             $X_k$;
            Augment ${\frac{\partial J}{\partial X}}^F[\cdot, k, l]$ by the result;
        end
    end
    Sum up the $(3 \times n_{\text{sh}} \times l_q^{\partial})$ tensor ${\frac{\partial J}{\partial X}}^F$ obtained over the quadratures;
    Augment $\frac{\partial J}{\partial X_{\text{loc}}}[\cdot, \cdot, K(F)]$ by the result;
end
```

Algorithm 3: Compute $\frac{\partial J}{\partial X_{\text{loc}}}$

The algorithm 3 performs an element-wise computation of the partial derivative $\frac{\partial J}{\partial X_{\text{loc}}}$. We use the algorithm 2 to assemble those local derivatives to obtain the total partial derivative $\frac{\partial J}{\partial X}$.

4.1.4 Partial Derivatives Calculation: Total Sensitivity

We return to the adjoint equation described in (4.7). This linear equation consists of the stiffness matrix B, the adjoint state vector Λ and the right hand side vector $\frac{\partial J}{\partial U}$. Explicit expressions for the equation described in (2.6) are to be evaluated in global form. Therefore, we consider the global degrees of freedom $U = (u_j)_{j \in \{1,\dots,N\}}$, $u_j \in \mathbb{R}^3$, and the node coordinates X, where it is understood that $u_j = 0$ if $X_j \in \partial\Omega_D$. We give the equation (2.6) in terms of global variables U via

$$B(X)U = F(X), \tag{4.32}$$

where

$$B(X)_{(j,r),(k,s)} = B(e_r\theta_j, e_s\theta_k) = \underbrace{\lambda \int\limits_\Omega \nabla_r \cdot \theta_j \nabla_s \cdot \theta_k \, dx}_{B_1(e_r\theta_j, e_s\theta_k)} + \underbrace{2\mu \int\limits_\Omega \varepsilon(e_r\theta_j) : \varepsilon(e_s\theta_k) \, dx}_{B_2(e_r\theta_j, e_s\theta_k)} \tag{4.33}$$

and

$$F(X)_{(j,r)} = F(e_r\theta_j) = \int\limits_\Omega f \cdot e_r\theta_j \, dx + \int\limits_{\partial\Omega_N} g \cdot e_r\theta_j \, dA, \tag{4.34}$$

with e_r, $r = 1, 2, 3$ the standard Basis on $\mathbb{R}^3$.

For industrial models with many design variables, the stiffness matrix is too large for direct solvers. We therefore use the commercial iterative solver ABAQUS to solve the adjoint equation where the partial derivative $\frac{\partial J}{\partial U}$ can be seen as pseudo force.
We now consider the total sensitivity as described in the equation (4.8).

$$\frac{dJ(X, U(X))}{dX} = \frac{\partial J(X, U)}{\partial X} - \Lambda^T \left(\frac{\partial B(X)}{\partial X} U - \frac{\partial F(X)}{\partial X} \right).$$

The evaluation of the total sensitivity requires the computation of the three terms: $\frac{\partial J(X,U)}{\partial X}$,

$\Lambda^T\left(\frac{\partial B(X)}{\partial X}\right)U$ and $\frac{\partial F(X)}{\partial X}$. The first term is calculated by algorithm 3. The quantity $\frac{\partial B(X)}{\partial X}$ in the second term is a tensor of dimension $(N\times 3)\times(N\times 3)\times(N\times 3)$ and contains two indices that originate from the partial derivative with respect to X and another four indices which are contracted with Λ and U. Taking advantage of the fact that this tensor is contracted with Λ and U, we avoid a memory consuming one-to-one storage in the main memory. Thus we start with computing the quantity $\Lambda^T\left(\frac{\partial B(X)}{\partial X}\right)U$.

By deriving the equation (4.33) we obtain

$$\frac{\partial B(X)_{(q,r),(k,s)}}{\partial X}=\frac{\partial B_1(e_r\theta_q,e_s\theta_k)}{\partial X_{ji}}+\frac{\partial B_2(e_r\theta_q,e_s\theta_k)}{\partial X_{ji}}.$$

Note that we consider in the next the partial derivatives of the element stiffness matrices. For the first partial derivative, one obtains

$$\begin{aligned}\frac{\partial B_1(e_r\theta_q,e_s\theta_k)}{\partial X_{ji}}=\lambda\sum_{K\in\widehat{j}^{-1}(j)_{idx}\cap\widehat{j}^{-1}(q)_1\cap\widehat{j}^{-1}(k)_{idx}}\sum_{l=1}^{l_q}\Bigg\{&\frac{\partial}{\partial X_{ji}}\big(\omega_{lK}\big)\nabla_r\theta_q(\xi_l^K)\nabla_s\theta_k(\xi_l^K)\\&+\omega_{lK}\frac{\partial}{\partial X_{ji}}\big(\nabla_r\theta_q(\xi_l^K)\big)\nabla_s\theta_k(\xi_l^K)\\&+\omega_{lK}\nabla_r\theta_q(\xi_l^K)\frac{\partial}{\partial X_{ji}}\big(\nabla_s\theta_k(\xi_l^K)\big)\Bigg\}.\end{aligned}\tag{4.35}$$

Here the notation $\widehat{j}^{-1}(j)_{idx}$ stands for the projection of the set $\widehat{j}^{-1}(j)$ to the idx-th component (the index of the element).

The three partial derivatives $\frac{\partial}{\partial X_{ji}}\big(\omega_{lK}\big)$, $\frac{\partial}{\partial X_{ji}}\big(\nabla_r\theta_q(\xi_l^K)\big)$ and $\frac{\partial}{\partial X_{ji}}\big(\nabla_s\theta_k(\xi_l^K)\big)$ have to be computed. We start with the first partial derivative and we obtain

$$\frac{\partial}{\partial X_{ji}}\big(\omega_{lK}\big)=\frac{\partial}{\partial X_{ji}}\big(\widehat{\omega}_l\det(J_K(\widehat{\xi}_l))\big)=\widehat{\omega}_l\frac{\partial}{\partial X_{ji}}\big(\det(J_K(\widehat{\xi}_l))\big).\tag{4.36}$$

For any invertible matrix A, we have $\frac{d}{d\alpha}\det(A) = \det(A)\,\mathrm{tr}\,(A^{-1}\frac{dA}{d\alpha})$, so we get

$$\frac{\partial}{\partial X_{ji}}(\omega_{lK}) = \widehat{\omega_l}\det\left(J_K(\widehat{\xi_l})\right)\ \mathrm{tr}\ \left((J_K(\widehat{\xi_l}))^{-1}\frac{\partial J_K(\widehat{\xi_l})}{\partial X_{ji}}\right). \tag{4.37}$$

The term $\frac{\partial J_K(\widehat{\xi_l})}{\partial X_{ji}}$ has been calculated in (4.31), where we have to replace $\widehat{\xi_l^F}$ with $\widehat{\xi_l}$. For the second partial derivative in $\frac{\partial B_1}{\partial X_{ji}}$ we have

$$\frac{\partial \nabla_r\theta_q}{\partial X_{ji}}(\xi_l^K) = \frac{\partial}{\partial X_{ji}}\left[\left(J_K(\hat{\xi}_l)^T\right)^{-1}\widehat{\nabla}\widehat{\theta}_{\widehat{j}_K^{-1}(q)}(\hat{\xi}_l)\right]_r. \tag{4.38}$$

By referring to the equations (4.29) and (4.31), we replace the surface quadrature point $\hat{\xi}_l^F$ by the volume quadrature point $\hat{\xi}_l$ to complete the computation. The third partial derivative in (4.35) is completely analogous to the second. For the partial derivative $\frac{\partial B_2}{\partial X}$ we obtain

$$\begin{aligned}\frac{\partial B_2(e_r\theta_q, e_s\theta_k)}{\partial X_{ji}} = 2\mu &\sum_{K\in\widehat{j}^{-1}(j)_1\cap\widehat{j}^{-1}(q)_1\cap\widehat{j}^{-1}(k)_1}\ \sum_{l=1}^{l_q}\\ &\left\{\frac{\partial}{\partial X_{ji}}(\omega_{lK})\varepsilon\left(e_r\theta_q(\xi_l^K)\right)) : \varepsilon\left(e_s\theta_k(\xi_l^K)\right)\right.\\ &+\omega_{lK}\frac{\partial}{\partial X_{ji}}(\varepsilon\left(e_r\theta_q(\xi_l^K)\right))) : \varepsilon\left(e_s\theta_k(\xi_l^K)\right)\\ &\left.+\omega_{lK}\varepsilon\left(e_r\theta_q(\xi_l^K)\right)) : \frac{\partial}{\partial X_{ji}}(\varepsilon\left(e_s\theta_k(\xi_l^K)\right))\right\}.\end{aligned} \tag{4.39}$$

The first term is calculated in (4.36). For the second term, we observe that the linear elastic strain tensor field is given by

$$\varepsilon(e_r\theta_q(\xi_l^K))_{ab} = \frac{1}{2}\left(\delta_{rb}\nabla_a\theta_q(\xi_l^K)) + \delta_{ra}\nabla_b\theta_q(\xi_l^K)\right), \tag{4.40}$$

and we refer to the argument following Eq. (4.38) to conclude the computation of (4.39). By deriving the equation (4.34) with respect to X we obtain

$$\frac{\partial F_{(q,r)}}{\partial X_{ji}} = \frac{\partial F^{\text{vol}}_{(q,r)}}{\partial X_{ji}} + \frac{\partial F^{\text{surf}}_{(q,r)}}{\partial X_{ji}}.$$

For the volume part we have

$$\begin{aligned} \frac{\partial F^{\text{vol}}_{(q,r)}}{\partial X_{ji}} = \sum_{K \in \widehat{j}^{-1}(j)_1 \cap \widehat{j}^{-1}(q)_1} \sum_{l=1}^{l_q} \Bigg\{ & \frac{\partial}{\partial X_{ji}} \big(\omega_{lK}\big) f_r(\xi_l^K) \theta_q(\xi_l^K) \\ & + \omega_{lK} \frac{\partial}{\partial X_{ji}} \big(f_r(\xi_l^K)\big) \theta_q(\xi_l^K) + \omega_{lK} f_r(\xi_l^K) \frac{\partial}{\partial X_{ji}} \big(\theta_q(\xi_l^K)\big) \Bigg\}. \end{aligned} \tag{4.41}$$

The partial derivative of the volume determinant has been calculated in (4.36). The shape function $\theta_q(\theta_l) = \hat{\theta}_{\widehat{j}_K^{-1}(q)}(\hat{\xi}_l)$ does not depend on the node X_{ji}, which implies that the term $\omega_{lK} f_r(\xi_l^K) \frac{\partial}{\partial X_{ji}} \big(\theta_q(\xi_l^K)\big)$ vanishes. If the volume force density f does not depend explicitly on the position X_{ji}, the term $\omega_{lK} \frac{\partial}{\partial X_{ji}} \big(f_r(\xi_l^K)\big) \theta_q(\xi_l^K)$ vanishes too.
Finally, we have to calculate the partial derivative of the surface loads

$$\begin{aligned} \frac{\partial F^{\text{surf}}_{(q,r)}}{\partial X_{ji}} = \sum_{\substack{F \in \mathcal{N}_h \\ K(F) \in \widehat{j}^{-1}(j)_1 \cap \widehat{j}^{-1}(q)_1}} \sum_{l=1}^{l_q^F} \Bigg\{ & \frac{\partial}{\partial X_{ji}} \big(\omega_{lF}\big) f_r(\xi_l^F) \theta_q(\xi_l^F) \\ & + \omega_{lF} \frac{\partial}{\partial X_{ji}} \big(f_r(\xi_l^F)\big) \theta_q(\xi_l^F) + \omega_{lF} f_r(\xi_l^F) \frac{\partial}{\partial X_{ji}} \big(\theta_q(\xi_l^F)\big) \Bigg\}. \end{aligned} \tag{4.42}$$

We use (4.21) to calculate the first term. The same reasoning as for the volume force is applied to compute the second and the third term. We present in the next the algorithm used to compute the total shape sensitivity:

```
Data: Same as in Algorithm 1 plus FE volume quadrature points and weights.
Result: A $N \times 3$ tensor containing $\frac{dJ}{dX}$
1 Use Algorithm 1 to obtain $\frac{\partial J}{\partial U}$;
2 Solve the adjoint equation (4.7) numerically using a standard FE solver;
3 Use Algorithm 5 below and $\Lambda$ to obtain $\Lambda^T \frac{\partial B}{\partial X} U$;
4 Use Algorithm 6 and Algorithm 7 below and $\Lambda$ to obtain $\Lambda^T \frac{\partial F}{\partial X}$;
5 Use Algorithm 3 to obtain $\frac{\partial J}{\partial X}$;
6 Add the tensors from step 4–5 with the proper signs to obtain $\frac{dJ}{dX}$ according to
   (4.8).
```

Algorithm 4: Compute the Shape Sensitivity $\frac{dJ}{dX}$

The following three algorithms are thus needed for the numerical calculation of the total shape gradient $\frac{dJ}{dX}$. Remark that by the implementation of the above algorithm, the storage of the $(3 \times n_{\mathrm{sh}}) \times (3 \times n_{\mathrm{sh}}) \times (3 \times n_{\mathrm{sh}}) \times N_{el}$ tensor $\frac{\partial B}{\partial X_{loc}}$ will be in many cases impossible due to constraints of the main memory, even if a sparse data format is used. Therefore, we contract the displacement tensor and the adjoint tensor with $\frac{\partial B}{\partial X_{loc}}$ during the local calculation.

We will note that the computation of $\Lambda^T \frac{\partial B}{\partial X_{\mathrm{loc}}} U$ performed by Algorithm 5 is local and have to be assembled to obtain the global term $\Lambda^T \frac{\partial B}{\partial X} U$. This assembling procedure is similar to the algorithm 2 and is omitted here. In order to increase the running time efficiency we use a parallel computation over the elements $K \in \mathcal{T}_h$.
The computation of the derivative of the volume force with respect to the X will be given by the next algorithm 6 for the case where the volume force density is independent of X. We will discuss the necessary adjustments in the case where volume loads, as e.g. centrifugal loads, depend on X in the context of concrete models in Section 4.2.4.

```
Data: Same as in Algorithm 4, but with volume quadrature points and weights.
The adjoint state $\Lambda$.
Result: A $3 \times n_{\text{sh}} \times N_{el}$ tensor containing $\Lambda^T \frac{\partial B}{\partial X_{\text{loc}}} U$
initialization $\Lambda^T \frac{\partial B}{\partial X_{\text{loc}}} U \leftarrow 0$;
for all elements $K \in \mathcal{T}_h$ do
    initialize a $3 \times n_{\text{sh}} \times 3 \times n_{\text{sh}} \times 3 \times n_{\text{sh}} \times l_q$ tensor ${\frac{\partial B}{\partial X_{\text{loc}}}}^K \leftarrow 0$;
    initialize a $3 \times n_{\text{sh}}$ tensor $\Lambda_{\text{loc}}^K \leftarrow 0$;
    initialize a $3 \times n_{\text{sh}}$ tensor $U_{\text{loc}}^K \leftarrow 0$;
    for all local degrees of freedom, $j = 1, \dots, n_{\text{sh}}$ do
        for all local degrees of freedom, $k = 1, \dots, n_{\text{sh}}$ do
            for all local degrees of freedom, $q = 1, \dots, n_{\text{sh}}$ do
                for all volume quadrature points $\hat{\xi}_l$, $l = 1, \dots, l_q$ do
                    Compute $\frac{\partial}{\partial X_j}\big(\omega_{lK}\big)$ using (4.36) ff.;
                    Compute $\frac{\partial}{\partial X_j}\big(\nabla_{[\cdot]}\theta_q(\widehat{\xi}_l)\big)$ and $\frac{\partial}{\partial X_j}\big(\nabla_{[\cdot]}\theta_k(\widehat{\xi}_l)\big)$ using (4.38);
                    Compute ${\frac{\partial B}{\partial X_{\text{loc}}}}^K[\cdot, j, \cdot, k, \cdot, q, l]$ using (4.35) and (4.39);
                    Assign $\Lambda_{\text{loc}}^K[\cdot, k] \leftarrow \Lambda[\cdot, \widehat{j}(K, k)]$;
                    Assign $U_{\text{loc}}^K[\cdot, q] \leftarrow U[\cdot, \widehat{j}(K, q)]$;
                end
            end
        end
        Multiply ${\frac{\partial B}{\partial X_{\text{loc}}}}^K[\cdot, j, \cdot, k, \cdot, q, l]$ with $\Lambda_{\text{loc}}^K[\cdot, k]$ and $U_{\text{loc}}^K[\cdot, q]$ and sum
        over $q, k$ (along with related $xyz$ indices) and quadrature index $l$;
        Store the result in $\Lambda^T \frac{\partial B}{\partial X_{\text{loc}}} U[\cdot, j, K]$;
    end
end
```

Algorithm 5: Compute $\Lambda^T \frac{\partial B}{\partial X_{\text{loc}}} U$

```
Data: Same as in Algorithm 1, but with volume quadrature points and weights
Volume force vector $f$ (eventually depending on $X$)
The adjoint state $\Lambda$.
Result: A $3 \times n_{\text{sh}} \times N_{el}$ tensor $\Lambda^T \frac{\partial F^{\text{vol}}}{\partial X_{\text{loc}}}$
initialization $\Lambda^T \frac{\partial F^{\text{vol}}}{\partial X_{\text{loc}}} \leftarrow 0$;
for all elements $K \in \mathcal{T}_h$ do
    initialize a $3 \times n_{\text{sh}} \times 3 \times n_{\text{sh}} \times l_q$ tensor $\frac{\partial F^{K\ \text{vol}}}{\partial X_{\text{loc}}} \leftarrow 0$;
    initialize a $3 \times n_{\text{sh}}$ tensor $\Lambda^K_{\text{loc}} \leftarrow 0$;
    for all local degrees of freedom, $j = 1, \dots, n_{\text{sh}}$ do
        for all local degrees of freedom, $k = 1, \dots, n_{\text{sh}}$ do
            for all volume quadrature points $\hat{\xi}_l$, $l = 1, \dots, l_q$ do
                Compute $\frac{\partial}{\partial X_j}\big(\omega_{lK}\big)$ using (4.36) ff.;
                Compute $\frac{\partial f_{[\cdot]}}{\partial X_j}(\xi_l)$ (model specific);
                Compute $\frac{\partial F^{K\ \text{vol}}}{\partial X^K_{\text{loc}}}[\cdot, j, \cdot, k, l]$ using (4.41) ;
                Assign $\Lambda^K_{\text{loc}}[\cdot, k] \leftarrow \Lambda[\cdot, \widehat{j}(K, k)]$;
            end
        end
        Multiply $\frac{\partial F^{K\ \text{vol}}}{\partial X_{\text{loc}}}[\cdot, j, \cdot, k, \cdot, q, l]$ with $\Lambda^K_{\text{loc}}[\cdot, k]$ and sum over $k$ (along with
        the related $xyz$ index) and quadrature index $l$;
        Store the result in $\Lambda^T \frac{\partial F^{\text{vol}}}{\partial X_{\text{loc}}}[\cdot, j, K]$;
    end
end
```

Algorithm 6: Compute $\Lambda^T \frac{\partial F^{\text{vol}}}{\partial X_{\text{loc}}}$

Similar to the last algorithm, an element-wise calculation and contraction with the adjoint state Λ is used to overcome the storage problems. The next algorithm performs the computations of the term $\Lambda^T \frac{\partial F^{\text{sur}}}{\partial X_{\text{loc}}}$. The assembly to the global $3 \times N$ matrices $\Lambda^T \frac{\partial F^{vol}}{\partial X}$ and $\Lambda^T \frac{\partial F^{sur}}{\partial X}$ is standard. We have thus provided all the necessary subalgorithms to Algorithm 4, which allows us an efficient calculation of the shape sensitivity.

```
Data: Same as in Algorithm 1
Surface force vector $g$ (eventually depending on $X$)
The adjoint state $\Lambda$.
Result: A $3 \times n_{\text{sh}} \times N^{\partial}_{el}$ tensor $\Lambda^T \frac{\partial F^{\text{surf}}}{\partial X_{\text{loc}}}$

initialization $\Lambda^T \frac{\partial F^{\text{surf}}}{\partial X_{\text{loc}}} \leftarrow 0$;
for all faces $F \in \mathcal{N}_h$ do
    initialize a $3 \times n_{\text{sh}} \times 3 \times n_{\text{sh}} \times l^F_q$ tensor $\frac{\partial F^{K \ \text{surf}}}{\partial X_{\text{loc}}} \leftarrow 0$;
    initialize a $3 \times n_{\text{sh}}$ tensor $\Lambda^K_{\text{loc}} \leftarrow 0$;
    for all local degrees of freedom, $j = 1, \ldots, n_{\text{sh}}$ do
        for all local degrees of freedom, $k = 1, \ldots, n_{\text{sh}}$ do
            for all surface quadrature points $\hat{\xi}^F_l$, $l = 1, \ldots, l^F_q$ do
                Compute $\frac{\partial}{\partial X^{K(F)}_j}\big(\omega_{lF}\big)$ using (4.21) ff.;
                Compute $\frac{\partial g_{[\cdot]}}{\partial X^{K(F)}_j}(\xi_l)$ (model specific);
                Compute $\frac{\partial F^{K \ \text{surf}}}{\partial X_{\text{loc}}}[\cdot, j, \cdot, k, l]$ using (4.42) ;
                Assign $\Lambda^K_{\text{loc}}[\cdot, k] \leftarrow \Lambda[\cdot, \widehat{j}(K(F), k)]$;
            end
        end
        Multiply $\frac{\partial F^{K \ \text{surf}}}{\partial X_{\text{loc}}}[\cdot, j, \cdot, k, \cdot, q, l]$ with $\Lambda^K_{\text{loc}}[\cdot, k]$ and sum over $k$ (along with
        the related $xyz$ index) and quadrature index $l$;
        Store the result in $\Lambda^T \frac{\partial F^{\text{surf}}}{\partial X_{\text{loc}}}[\cdot, j, F]$;
    end
end
```

Algorithm 7: Compute $\Lambda^T \frac{\partial F^{\text{surf}}}{\partial X_{\text{loc}}}$

4.2 Numerical Examples and Validation

In this section we describe in more detail the implementation of the total sensitivity by using the adjoint method. Later we compute the form gradient for the probability of failure for three different mechanical models with increasing geometric and numeric complexity.

4.2.1 Implementation Details

The implementation of the partial derivatives $\frac{\partial J}{\partial U}$ and $\frac{\partial J}{\partial X}$ is based on scripts in R 3.2.1. We use python scripts and the commercial solver ABAQUS CAE 6.13 to compute the adjoint equation. The computation of the total sensitivity is carried out in parallel on a different cores using the R-package parallel. In our case, the maximal number of parallel computing cores is 12. We use the R-package tensorA for some tensor summations and contractions.

Element types available are linear or quadratic rectangular plane stress elements, linear or quadratic triangular plane stress elements, tetrahedra elements with 4 or 8 DoF and brick elements with 8 or 20 DoF. Quadratures for the surface integrals can be chosen with up to 81 quadrature points, while full and reduced quadratures are available for volume integrals.

The boundary conditions such clamped Dirichlet and cyclic boundary conditions have been implemented along with volume forces and surface forces.

4.2.2 A 2D Bended Rod under Tensile Loading

We consider as a first example a simple two dimensional plane stress model where we have some intuitive prediction about the behavior of the probabilistic shape gradient. We thus consider a geometry of a rod which is 100 *mm* long and is bended up to a width of 25 *mm*. We use the same material properties as described in Section 3.4.2. A Dirichlet boundary condition $u = 0$ is applied on the left and a uniform tensile stress load is applied on the right with an amplitude of 12 N/mm^2 .

Mesh Convergence Study

The accuracy of the numerical solution by using the finite element method depends on the number of mesh elements. However, the computational cost increases with each successive level of mesh refinement. We perform a mesh convergence study to find a solution that balance the accuracy of the numerical solution and the computational capacity in a reasonable way.

We start by refining successively the mesh of our model with denser element distribution and we compute the probabilistic cost functional and the Weibull scale η for each mesh refinement step (see table 4.1).

Mesh density	Elements	Coordinates	Faces	Probabilistic functional	Weibull scale
Mesh 1	265	912	116	1.28e-03	27.9
Mesh 2	1060	3413	232	3.50e-03	16.8
Mesh 3	2114	6673	330	4.60e-03	14.7
Mesh 4	4220	13123	462	5.63e-03	13.3
Mesh 5	7228	22293	608	6.27e-03	12.6
Mesh 6	13572	41543	826	6.93e-03	12.0
Mesh 7	26100	79445	1144	7.44e-03	11.5
Mesh 8	47101	142844	1540	7.80e-03	11.3
Mesh 9	53534	162253	1650	7.86e-03	11.2
Mesh 10	62524	189351	1778	7.94e-03	11.2

Table 4.1: Mesh convergence for the probabilistic cost functional $J(\Omega, u)$ and the Weibulls scale η

Figure 5-3 shows the converging trend of the Weibull scale variable η as mesh density increases. To reduce the computational time by an acceptable accuracy the mesh 5 are selected for this model.

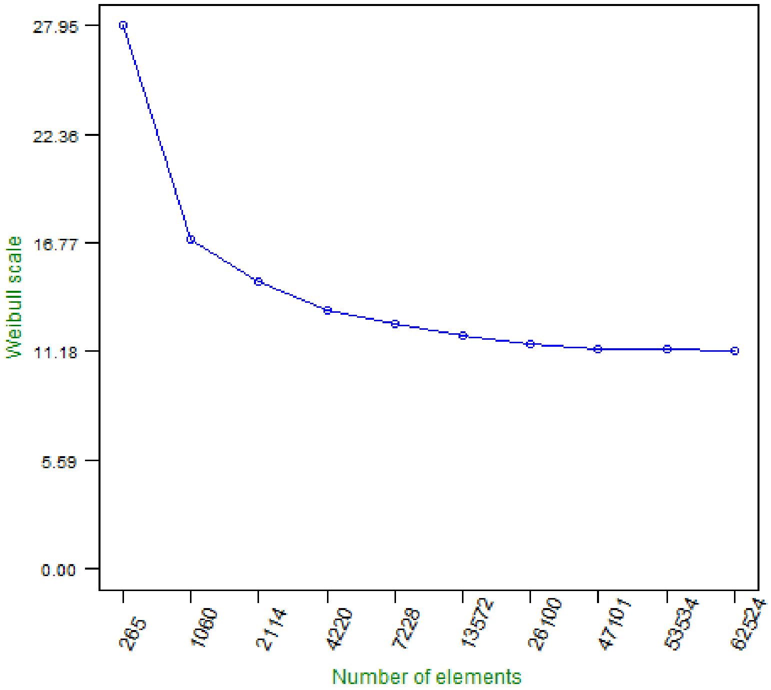
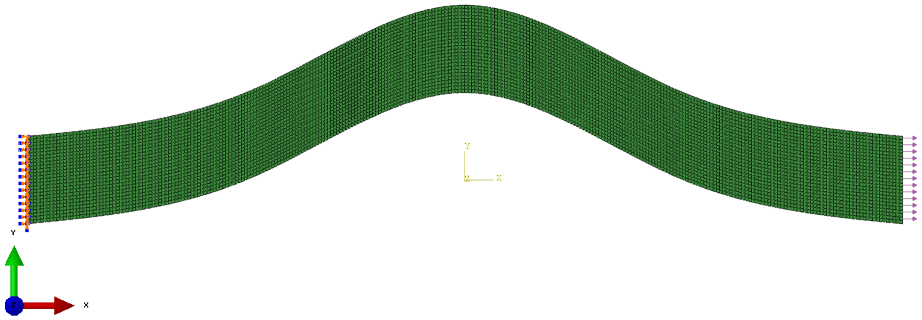


Figure 4-1: The mesh convergence for Weibull scales.

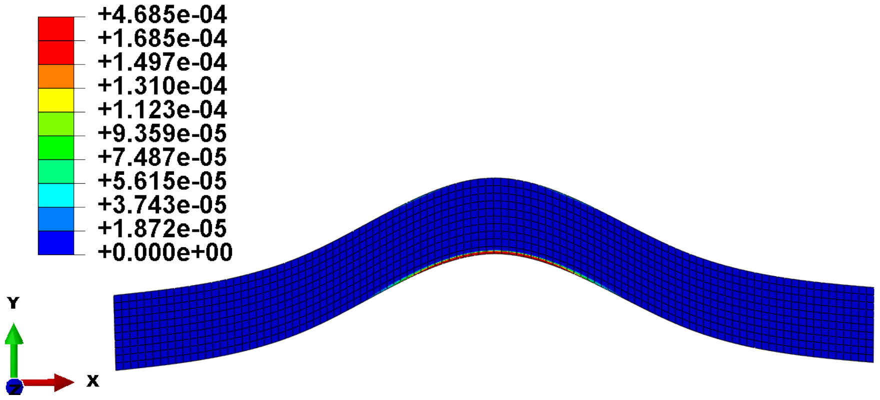
Finite Element Model

The finite element model consists out of 22293 nodes and 7228 rectangular quadratic stress elements of type CPS8 with $n_{sh} = 8$ local degrees of freedom. The volume quadrature contains $l_q = 9$ points and the surface quadrature contains $l_q^F = 4$ points. The value of the objective functional J is 6.27×10^{-3}, which corresponds to a Weibull scale variable $\eta = 12.62$ cycles.

Figure 4-2a shows the 2D bended rod model subjected to a surface traction load on the right side where the left side is clamped ($x = y = z = ux = uy = uz = 0$). Figure 4-2b shows the crack formation intensity due to the uniform tensile stress. We show that the critical region (red hotspot) for crack initiation is located at the lower side of the portion of the rod with the strongest bending.

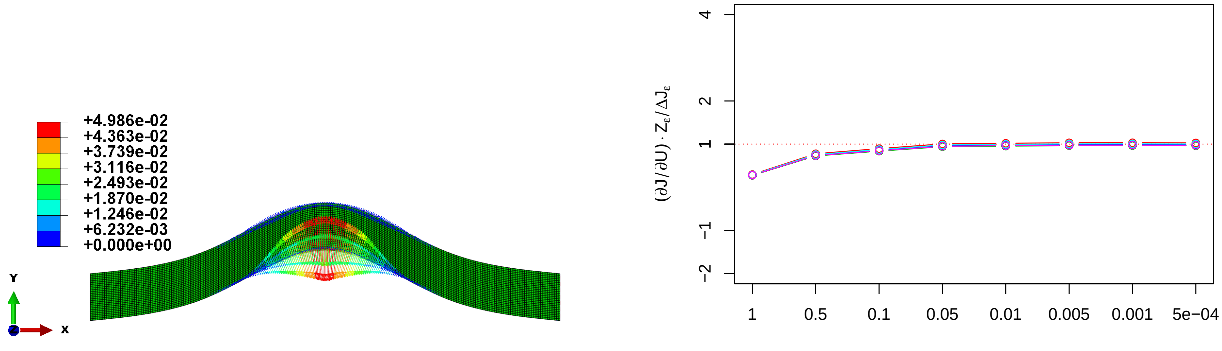


(a) The bended rod subjected to surface traction force on the right side. The left side is clamped with encastre boundary conditions



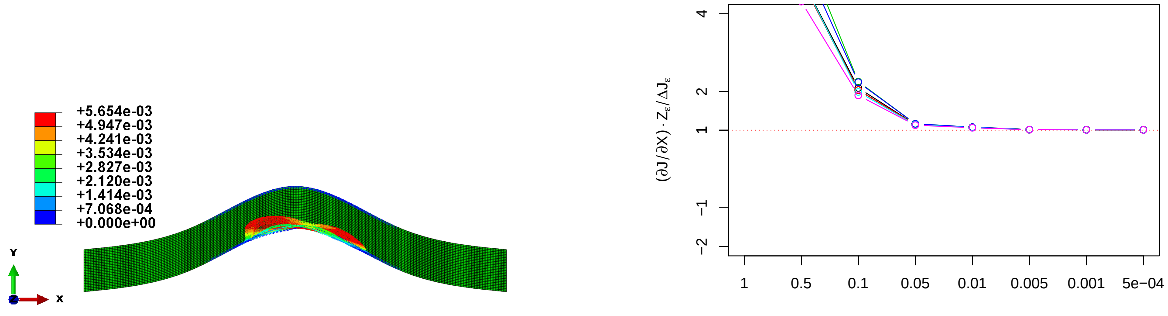
(b) Crack formation intensity for the bended rod. Stress concentration on the lower side of the location with the strongest curvature leads to an augmented probability of stress initiation.

Figure 4-2: 2-D bended rod under tensile loading and Crack formation intensity



(a) Partial derivative $\frac{\partial J}{\partial U}_j$ visualized as an arrow at node X_j.

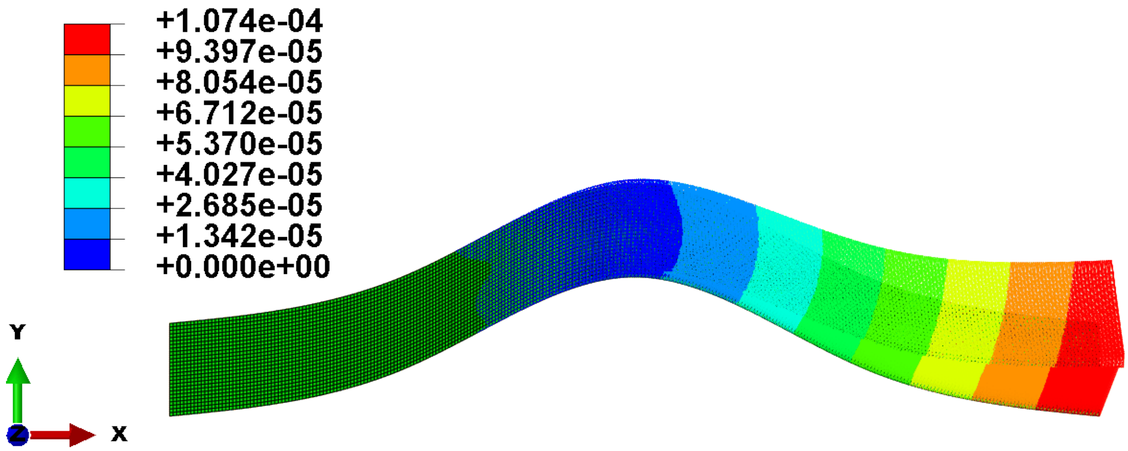
(b) Comparison of $\frac{\partial J}{\partial U} \cdot Z_i$ with the finite differences $\frac{J(X,U+\varepsilon Z_i)-J(X,U)}{\varepsilon}$ for $\varepsilon \to 0$. Z_i is a random perturbation field ($i = 6$).



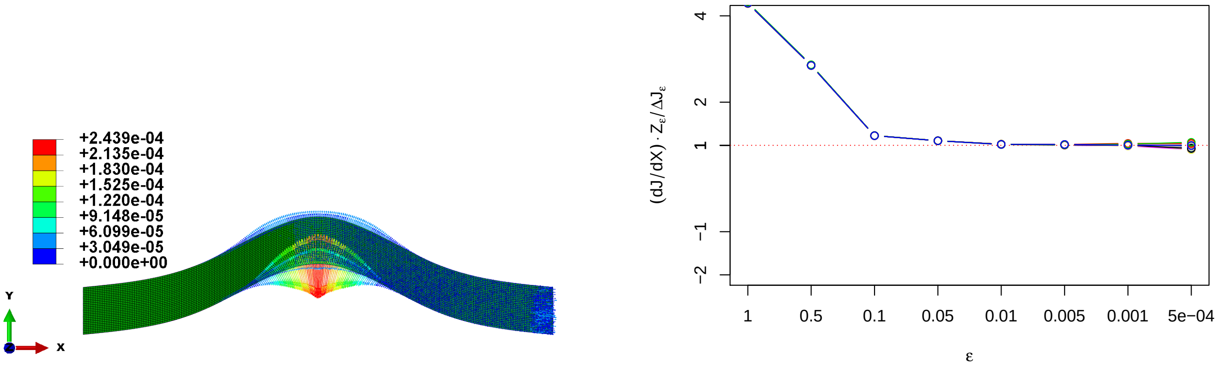
(c) The jth component of $\frac{\partial J}{\partial X}$ is visualized as an arrow at node X_j.

(d) Comparison of $\frac{\partial J}{\partial X} \cdot Z_i$ with the finite differences $\frac{J(X+\varepsilon Z_i,U)-J(X,U)}{\varepsilon}$ for $\varepsilon \to 0$. Z_i is a random perturbation field ($i = 6$).

Figure 4-3: Partial derivatives of J.



(a) Adjoint state Λ_j is visualized as an arrow at X_j. Asymmetry is a consequence of non symmetric boundary conditions.



(b) The total shape sensitivity $\frac{dJ}{dX}$. The jth component of $\frac{dJ}{dX}$ is visualized an arrow at node X_j.

(c) Comparison of the shape sensitivity $\frac{dJ}{\partial X} \cdot Z_i$ with the finite differences $\frac{J(X+\varepsilon Z_i, U(X+\varepsilon Z_i)) - J(X,U)}{\varepsilon}$ for $\varepsilon \rightarrow 0$. Z_i is a random perturbation field.

Figure 4-4: Validation of the shape sensitivities.

Figures 4-3a and 4-3c show the partial derivatives $\frac{\partial J}{\partial U}$ and $\frac{\partial J}{\partial X}$ as arrows at mesh nodes. The panels in the figures 4-3b and 4-3d compare results of the partial derivatives with the finite difference method. Note that the finite difference method is calculated based on a small random perturbation of the mesh geometry. The range of relative errors by the numerical validation is 0.1%.

In figure 4-4a we show the adjoint state. Note that the asymmetry of the boundary conditions imposed only on the left hand side yields an appearance of asymmetry of the adjoint state field. The total shape sensitivity is displayed in Figure 4-4b. The panel in figure 4-4c gives the validation of the total sensitivity and the finite difference method by random perturbations on the mesh coordinates. The relative error by the numerical validations is less than 1%.

The total shape sensitivity tends to enlarge the structure of the bended rod at the area of the highest loading and will thus increase the diameter at this location, which will effectively decrease the peak stress and thereby the failure probabilities.

4.2.3 A 3D Bended Rod under Tensile Loading

As a second test, we consider a three dimensional bended rod subjected to a uniform surface traction loading. We thus consider a geometry of a 1 mm diameter rod which is 6.08 mm long, bended up to a height of 3.51 mm. We use the same material properties as in Section 3.4.3. A Dirichlet boundary condition $u = 0$ is applied on the fixed left side and a uniform surface traction force is applied on the right side with an amplitude of 12 N/mm^2.

Mesh Convergence

We perform a finite element discretization of the 3D bended rod by using a quadratic hexahedral element. By increasing successively the mesh density we calculate for every mesh the probabilistic cost functional. Table 4.2 summarizes the mesh densities information and the corresponding obtained Weibull scales.

Mesh density	Elements	Coordinates	Faces	Probabilistic functional	Weibull scale η	Relative Dev. η %
Mesh 1	570	2950	398	1.36e-11	271163	7.6
Mesh 2	918	4550	530	1.20e-11	288675	1.7
Mesh 3	1302	6410	734	1.26e-11	281718	4.1
Mesh 4	7700	34750	2510	1.35e-11	272165	7.3
Mesh 5	12816	56448	3314	1.15e-11	294884	0.4
Mesh 6	18824	82328	4522	1.22e-11	286299	2.5
Mesh 7	25520	110673	5544	1.19e-11	289885	1.3
Mesh 8	31330	135712	6722	1.09e-11	302891	3.2
Mesh 9	42582	182567	7936	1.16e-11	293610	0.0
Mesh 10	56610	241226	9614	1.16e-11	293610	—

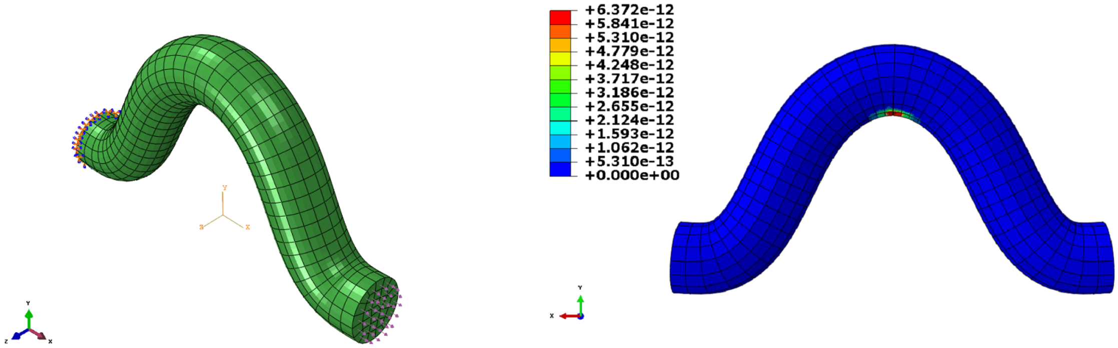
Table 4.2: Mesh convergence for the probabilistic cost functional $J(\Omega, u)$ and the Weibulls cale η.

Taking the finest mesh as the reference, the maximum deviation is well below 10% with a trend towards smaller deviation for growing mesh size. A certain (declining) level of meshing noise is observed, which however is in the same range as maximum stress variations in deterministic life calculation. In order to balance the accuracy and the simulation efficiency the mesh 3 is used.

Finite Element Model

The finite element mesh consists out of 6410 nodes and 1302 quadratic brick elements of type C3D20R with $n_{sh} = 20$ local degrees of freedom. The reduced volume quadrature contains $l_q = 8$ points and the surface quadrature contains $l_q^F = 36$ points. The value of the objective functional J is 3.998054×10^{-11}, which corresponds to a Weibull scale variable $\eta = 158152$ cycles.

The applied boundary conditions and load vector are illustrated in Figure 4-5a. As expected, during the load process, the crack formation intensity is higher at the lower side of the critical region with the strongest bending, see Figure 4-5b.



(a) The bended rod subjected to surface traction force on the right side. The left side is clamped with encastre boundary conditions

(b) Crack formation intensity for the bended rod. Stress concentration on the lower side of the location with the strongest curvature leads to an augmented probability of stress initiation.

Figure 4-5: 3-D bended rod under tensile loading and Crack formation intensity

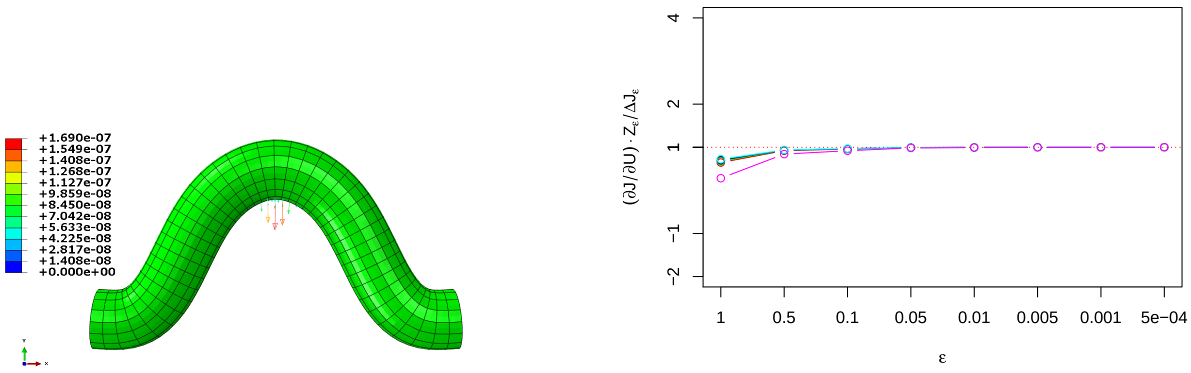
Figures 4-6a and 4-6c show the partial derivatives of the objective functional with respect to the displacements and coordinates. Both partial derivatives $\frac{\partial J}{\partial U}$ and $\frac{\partial J}{\partial X}$ can be obtained in direction Z by the following equations:

$$Z_i^u \cdot \frac{\partial J}{\partial U} = \lim_{\epsilon \to 0} \frac{J(X, U + \epsilon Z_i^u) - J(X, U)}{\epsilon}$$

and

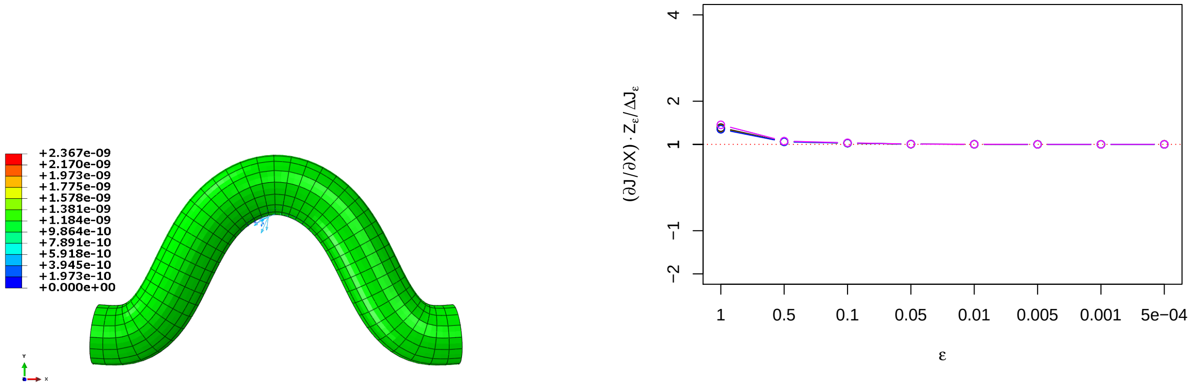
$$Z_i^x \cdot \frac{\partial J}{\partial X} = \lim_{\epsilon \to 0} \frac{J(X + \epsilon Z_i^x, U) - J(X, U)}{\epsilon}$$

where Z_i^u and Z_i^x are small random perturbations of the displacements and coordinates field respectively.



(a) Partial derivative $-\frac{\partial J}{\partial U}_j$ visualized as an arrow at node X_j. Rescaling is applied for better visualization.

(b) Comparison of $\frac{\partial J}{\partial U} \cdot Z_i$ with the finite differences $\frac{J(X,U+\varepsilon Z_i)-J(XV,U)}{\varepsilon}$ for $\varepsilon \to 0$. Z_i is a random perturbation field ($i = 6$). All six tests show good agreement, as the quotient of the shape sensitivity divided by finite differences approaches one for small ε.



(c) The jth component of $\frac{\partial J}{\partial X}$ is visualized an arrow at node X_j.

(d) Comparison of $\frac{\partial J}{\partial X} \cdot Z_i$ with the finite differences $\frac{J(X+\varepsilon Z_i,U)-J(X,U)}{\varepsilon}$ for $\varepsilon \to 0$. Z_i is a random perturbation field ($i = 6$). All six tests show good agreement, as the quotient of the shape sensitivity divided by finite differences approaches one for small ε.

Figure 4-6: Partial derivatives of J.

By assuming that the random perturbations Z_i^u and Z_i^x are small enough that the linear approximations are sufficiently accurate, we calculate the euclidian norm of $\| \frac{\partial J}{\partial U} \cdot Z_i^u \| / \| \frac{J(X,U+\varepsilon Z_i^u)-J(X,U)}{\varepsilon} \|$ and the euclidian norm of $\| \frac{\partial J}{\partial X} \cdot Z_i^x \| / \| \frac{J(X+\varepsilon Z_i^x,U)-J(X,U)}{\varepsilon} \|$ with decreasing ε and various random perturbations Z_i^u and Z_i^x with $i = 1, ..., 6$. see figures 4-6b and 4-6d.

For the total shape sensitivity we can write the following finite difference approximation:

$$Z_i^x . \frac{dJ}{dX} \approx \frac{J(X + \epsilon Z_i^x, U(X + \epsilon Z_i^x)) - J(X, U)}{\epsilon}$$

Note that for every random perturbation Z_i^x $(i = 1, ..., 6)$, we have to run the FE solver to compute the new displacement field $U(X + \epsilon Z_i^x)$ correspond to the new geometry $X + \epsilon Z_i^x$. A comparison between the total shape sensitivity and the finite difference method for six random geometry perturbations is shown in figure 4-8b. Figure 4-7 visualizes the total shape gradient. It is visible that the arrows point outward from the critical region in the normal directions which approves our intuitive prediction.

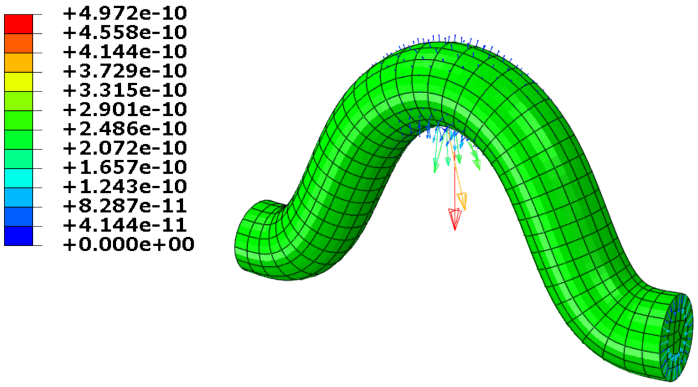
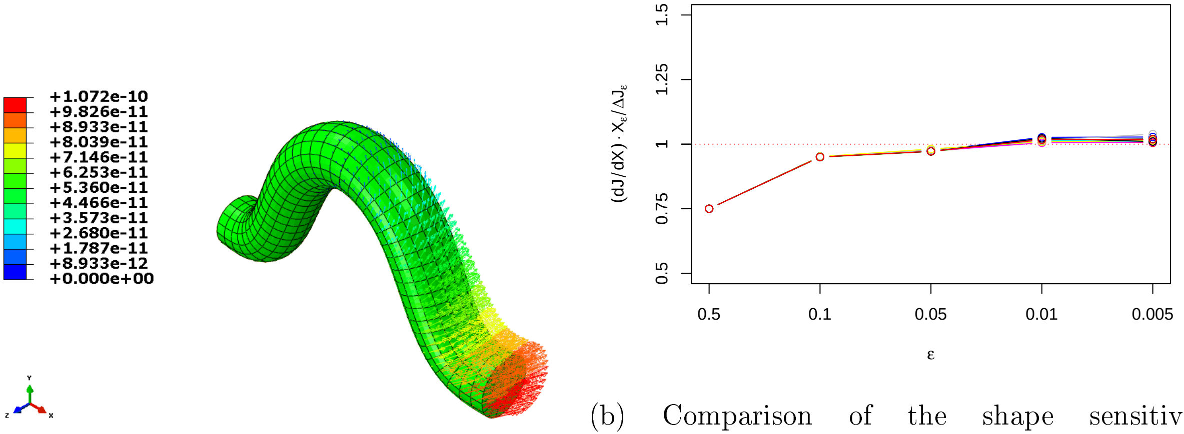


Figure 4-7: The total shape sensitivity $\frac{dJ}{dX}$. The jth component of $-\frac{dJ}{dX}$ is visualized an arrow at node X_j.



(a) Adjoint state Λ_j is visualized as an arrow at X_j. Asymmetry is a consequence of non symmetric boundary conditions. Rescaling is applied for better visualization.

(b) Comparison of the shape sensitivity $\frac{dJ}{\partial X} \cdot Z_i$ with the finite differences $\frac{J(X+\varepsilon Z_i, U(X+\varepsilon Z_i))-J(X,U)}{\varepsilon}$ for $\varepsilon \to 0$. Curves for six different random directions Z_i which are applied to deform the mesh X are chosen and displayed in different colors (but partially mask each other due to almost identical behavior). All six tests show good agreement, as the quotient of the shape sensitivity divided by finite differences approaches one for small ε.

Figure 4-8: Adjoint state and validation of the shape sensitivities.

The Table 4.3 reports execution times for the computation steps of shape sensitivity for the bended rod. The leading computational cost lies in the calculation of partial derivatives of the stiffness matrix $\frac{\partial B}{\partial X}$, cf. Algorithm 5. For the C3D20R brick elements used here, the element wise tensor representing this expression contains $(3 \times 20)^3 = 216\ 000$ entries, which have to be computed for each element before contracting with Λ and U. Fortunately, this task can be easily carried out in parallel. Usage of more than 6 virtual cores will thus considerable shorten the execution time.

Quantity	**Elapsed**	**Cores**	**Tool**
State U	29.81	1	`ABAQUS CAE 6.13`
$\frac{\partial J}{\partial U}, \frac{\partial J}{\partial X}$ together	77.22	1	`R 3.1.0`
Adjoint State Λ	29.67	1	`ABAQUS CAE 6.13`
$\frac{\partial B}{\partial X}, \frac{\partial F}{\partial X}$ together	154.51	6	`R 3.1.0`

Table 4.3: Execution times in sec for the bended rod model on an Intel Core i7-3632QM CPU @ 2.20GHZ, 12GB shared memory machine with 4 physical and 8 virtual cores.

Also from a mechanical standpoint, the results are convincing. While the negative shape sensitivities of the probability of failure all point outwards – more material brings more reliability – the highest sensitivities are observed at the location of stress concentration at the lower bottom of the rod's bow. A shape flow following the direction of the negative shape gradient would thus straighten out the rod to improve it's reliability, just as general mechanical wisdom suggests.

Execution times for the computation steps of shape sensitivity for the bended rod are reported in Table 4.3. We see that the calculation of the partial derivatives of the stiffness matrix is the most time-consuming step. This can be explained by the fact that even single element shape derivatives of the stiffness matrix with 20 DoF and 8 volume quadrature points for a reduced quadrature requires the calculation of an array of dimension $20 \times 3 \times 20 \times 3 \times 20 \times 3 \times 8$ which is more than 1.728 million array entries per element. Reducing the number of degrees of freedom lead to much more noisy representations of the shape sensitivities.

4.2.4 A Turbo Charger Compressor of a Jet Engine

We next calculate the total shape sensitivity for a radial turbo compressor of a jet engine introduced in Section 3.4.3. The load vector, neglecting gas pressure, is exclusively determined by the centrifugal load and the density $\rho = 2.65 \times 10^{-9} kg/mm^3$. The rotation speed is 110000 rpm and corresponds to $\omega = (2\pi \times 110000)/60 = 11519.17 Hz$.
The centrifugal force points radially outward from the axis of rotation x and can be expressed as $f = \rho\omega\alpha$, where $\alpha = (0, \alpha_2, \alpha_3)$. Then, the derivative of the centrifugal force with respect to the mesh nodes can be written as:

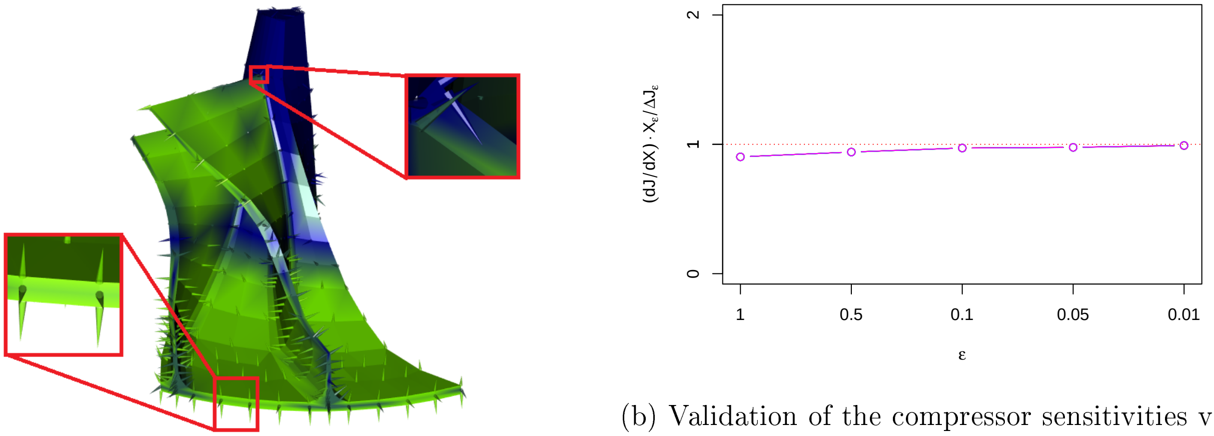
$$\frac{\partial f}{\partial X_{ji}}(\xi_l^K) = \rho\omega^2 \frac{\partial}{\partial X_{ji}} T_K(\hat{\xi}_l) = \begin{cases} 0, & \text{if } i = 1; \\ \rho\omega^2 \hat{\theta}_j(\hat{\xi}_l) e_i, & \text{if } i = 2, 3. \end{cases}$$

where X_{ji} is the local node coordinate.
We remember that the compressor consist of 7 symmetric sectors. Therefore, we profit from this cyclic symmetry structure by analyzing the shape sensitivity only for a single repetitive sector instead of the entire compressor. Thus the computational cost is considerably reduced. The total sensitivity for all segments can be evaluated by the following relation:

$$\left[\frac{dJ}{dX}\right]_{k+1} = R_x(\theta). \left[\frac{dJ}{dX}\right]_k$$

where $R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{pmatrix}$ is the rotation matrix about the x-axis by an angle $\theta = 2\pi/7$ and $\left[\frac{dJ}{dX}\right]_k$ stands for sensitivities of sector k $(k = 1, ..., 6)$.

The numerical simulation is performed with the settings given in Section 3.4.3. The details for the finite element simulation are given in 3.3. Note that we apply the same



(a) Positive shape sensitivity for the radial compressor (only a section of 1/7 th of the Geometry in Fig. 3-4 is displayed). Length of $\frac{dJ}{dX}_j$ is displayed as arrow at X_j (rescaled). The left clip shows a zoom of positive sensitivities, while the right clip zooms negative sensitivities.

(b) Validation of the compressor sensitivities vs finite differences. Curves for six different random directions which are applied to deform the mesh X are chosen and displayed in different colors (but mask each other due to almost identical behavior). All six tests show good agreement, as the quotient of the shape sensitivity divided by finite differences approaches one for small ε.

Figure 4-9: Compressor shape sensitivity and validation test .

discretization for the adjoint state. Table 4.4 shows runtime information. Again, as already explained in section 4.2.3, the leading computation time is spent in algorithm 5 and can be considerably reduced, when parallelizing with more cores.

Figure 4-9 displays the shape sensitivity for the radial compressor. Also in the case of a complex geometry the shape sensitivity calculations conform to more than 99% with the finite difference method.

As in the case of the bended rod, the physical interpretation of the results is well aligned with general mechanical insight. Due to centrifugal load, when targeting reliability, the worst design option is adding more material at far from the rotation axis. Therefore the *positive* shape sensitivities, pointing to worsened reliability, are large and outward pointing in such locations, see the left clip in Figure 4-9 (a). Despite the additional centrifugal load, it is however favorable to reliability to add more material in the fillet

Quantity	**Elapsed**	**Cores**	**Tool**
State U	45.95	1	`ABAQUS CAE 6.13`
$\frac{\partial J}{\partial U}, \frac{\partial J}{\partial X}$ together	71.77	1	`R 3.1.0`
Adjoint State Λ	53.49	1	`ABAQUS CAE 6.13`
$\frac{\partial B}{\partial X}, \frac{\partial F}{\partial X}$ together	195.91	6	`R 3.1.0`

Table 4.4: Execution times for the compressor model in sec on an Intel Core i7-3632QM CPU @ 2.20GHZ, 12GB shared memory machine with 4 physical and 8 virtual cores.

of the compressor's blade close to the rotation axis, as is shown by the outward pointing *negative* sensitivities in the right clip in the same Figure.

Chapter 5

Thermo-Elastic Sensitivity Analysis

In many engineering fields, like gas turbine design, thermo-elastic effects caused by the change of temperature on the state of materials have to be considered. There are various approaches for modelling such effects by coupling temperature and stress [54, 55]. By fully coupled thermal-stress analysis, stress field and temperature distribution affect each other strongly [56]. In our study, we use the sequentially coupled approach, which assume that strains arising from boundary loadings and body forces induce small temperature changes and hence can be neglected (which is the case in gas turbine blades). This approximate, sequentially coupled approach is satisfactory for materials like metals (see [54]).

In the first subsection, the linear thermoelasticity problem composed of two partial differential equations, describing the mechanical and thermal behavior, is introduced. Furthermore, we present the weak formulation and an algebraic form of the coupled system. A temperature dependent probabilistic LCF-model is described in section 5.4. Expressions for computing the total thermo-elastic sensitivity using adjoint method are derived in section 5.5. Finally, we present some numerical test cases on 2D and 3D geometries.

5.1 Linear Thermoelasticity Problem

The objective of our thermoelastic analysis is to compute the shape sensitivity of components subject to mechanical and thermal loadings. Within the context of linear small deformation theory, the total strain can be decomposed into the sum of mechanical and thermal components, see [27]:

$$\varepsilon_{ij} = \varepsilon_{ij}^{M} + \varepsilon_{ij}^{T}.$$

If T_0 is taken as the reference temperature and T as an arbitrary temperature, the thermal strain due to the change of temperature is:

$$\varepsilon_{ij}^{T} = \alpha_{ij}(T - T_0)$$

where $T - T_0$ is the temperature difference and α_{ij} are coefficients of the thermal expansion tensor. If the material is taken as isotropic, then α_{ij} must be an isotropic second-order tensor, and we have

$$\varepsilon_{ij}^{T} = \alpha(T - T_0)\delta_{ij}.$$

The stationary heat equation with constant thermal conductivity and without a heat source within the volume is simply the Laplace equation [27]:

$$\triangle T = 0,$$

where $\triangle$ is the Laplace operator. The heat flux transfer rate is given by:

$$q = -k\nabla T,$$

where k is the thermal conductivity constant.

In case of convective heat transfer, the heat flux $n \cdot q = k\frac{\partial T}{\partial n}$ over the boundary is proportional to the difference of the temperature $T - T_{ext}$. Note that the temperature gradient is in a direction of greatest decrease of temperature and hence the opposite direction of the outward pointing normal vector of the boundary. Thus the heat flux will have a positive sign if the external temperature is lower than the internal temperature and a negative sign otherwise.

For a positive heat flux coefficient $h > 0$ we have the dependence

$$n \cdot q = h(T - T_{ext}).$$

The temperature distribution satisfies the following equation:

$$\begin{cases} \Delta T = 0, & \text{in } \Omega; \\ -k\frac{\partial T}{\partial n} = h(T - T_{ext}), & \text{on } \partial\Omega. \end{cases}$$

The governing equation of linear isotropic thermoelasticity reads:

$$\begin{cases} \nabla \cdot \sigma + f = 0, & \text{in } \Omega; \\ \sigma(u) = \mu(\nabla u + \nabla u^T) + & \\ \quad [\lambda(\nabla \cdot u) - \alpha(3\lambda + 2\mu)(T - T_0)]\mathfrak{I}, & \text{in } \Omega; \\ u = 0, & \text{on } \partial\Omega_D, \\ \sigma(u) \cdot n = g, & \text{on } \partial\Omega_N. \end{cases} \tag{5.1}$$

We note that the component is clamped on the Dirichlet boundary $\partial\Omega_D$ and a surface traction force g is applied on the Neumann boundary $\partial\Omega_N$.

5.2 Weak Formulation

In order to use finite element approximations, we first have to transform the partial differential equations to the weak formulation. Thereafter, a discretization of the derived linear and bilinear forms is required.

Applying the weak formulation on the heat equation yields

$$\int\limits_{\Omega} \Delta T v_T \; dx = \int\limits_{\partial\Omega} (\nabla T \cdot n) v_T \; dA - \int\limits_{\Omega} \nabla T \cdot \nabla v_T \; dx = 0,$$

where $v_T \in H^1(\Omega)$ is a testfunction. It follows that

$$-\frac{h}{k} \int\limits_{\partial\Omega} (T - T_{ext}) v_T \; dA = \int\limits_{\Omega} \nabla T \cdot \nabla v_T \; dx.$$

We reorder the last equation by placing the solution T of the heat transfer equation on the left side. Thus we have

$$\int\limits_{\Omega} k \nabla T \cdot \nabla v_T \; dx + \int\limits_{\partial\Omega} h T v_T \; dA = \int\limits_{\partial\Omega} h T_{ext} v_T \; dA.$$

On the other side, we write the weak formulation of the linear thermoelasticity as

$$-\int\limits_{\Omega} \nabla\sigma(u) \cdot v_M \; dx = \int\limits_{\Omega} \overbrace{\sigma(u) : \nabla v_M}^{\sigma(u):\varepsilon(v_M)} \; dx - \int\limits_{\partial\Omega} v_M \overbrace{\sigma(u) \cdot n}^{g} \; dA = \int\limits_{\Omega} f \cdot v_M \; dx,$$

where $v_M \in V_{DN} = \{v \in [H^1(\Omega)]^3; v = 0 \text{ on } \partial\Omega_D\}$.

Coupling the weak formulations of the linear thermoelasticity and heat transfer gives the following system of equations:

$$\begin{cases} \int\limits_{\Omega} \sigma(u) : \varepsilon(v_M)\ dx = \int\limits_{\partial\Omega} g v_M dA + \int\limits_{\Omega} f v_M\ dx \\ \int\limits_{\Omega} \nabla T \cdot \nabla v_T\ dx = -\frac{h}{k} \int\limits_{\partial\Omega} (T - T_{ext}) v_T\ dA \end{cases} \tag{5.2}$$

By substituting the global stress term of the governing equation (5.1) into the first term in the system of equation (5.2) we obtain

$$\begin{aligned} \int\limits_{\Omega} \sigma(u) : \varepsilon(v_M)\ dx =& \lambda \int\limits_{\Omega} \nabla \cdot u \nabla \cdot v_M\ dx + 2\mu \int\limits_{\Omega} \varepsilon(u) : \varepsilon(v_M)\ dx - \\ & \alpha(3\lambda + 2\mu) \int\limits_{\Omega} T \cdot \operatorname{tr}(\varepsilon(v_M))\ dx + \alpha(3\lambda + 2\mu) \int\limits_{\Omega} T_0 \cdot \operatorname{tr}(\varepsilon(v_M))\ dx. \end{aligned}$$

The weak formulation of the thermo-mechanical equation gives

$$\begin{aligned} & \lambda \int\limits_{\Omega} \nabla \cdot u \nabla \cdot v_M\ dx + 2\mu \int\limits_{\Omega} \varepsilon(u) : \varepsilon(v_M)\ dx - \alpha(3\lambda + 2\mu) \int\limits_{\Omega} T \cdot \operatorname{tr}(\varepsilon(v_M))\ dx \\ & = -\alpha(3\lambda + 2\mu) \int\limits_{\Omega} T_0 \cdot \operatorname{tr}(\varepsilon(v_M))\ dx + \int\limits_{\partial\Omega} g v_M\ dA + \int\limits_{\Omega} f v_M\ dx. \end{aligned}$$

5.3 Discretized Thermo-mechanical System of Equations

In this section, we represent the finite element discretization of the coupled thermo-mechanical system. The method is analog to the purely mechanical case described in the last chapter. In this section, we concentrate on the discretization of the thermal terms. First, we define the following bilinear forms and linear forms. The purely mechanical bilinear form is defined as:

$$B_M(u, v_M) = \lambda \int\limits_{\Omega} \nabla \cdot u \nabla \cdot v_M \; dx + 2\mu \int\limits_{\Omega} \varepsilon(u) : \varepsilon(v_M) \; dx.$$

The thermal-mechanical bilinear form reads

$$B_{MT}(T, v_M) = -\alpha(3\lambda + 2\mu) \int\limits_{\Omega} T \cdot \operatorname{tr}(\varepsilon(v_M)) \; dx.$$

The thermal bilinear form is

$$B_T(T, v_T) = k \int\limits_{\Omega} \nabla T \cdot \nabla v_T \; dx + h \int\limits_{\partial\Omega} T v_T \; dA.$$

The mechanical linear form is defined as

$$F_M(v_M) = -\alpha(3\lambda + 2\mu) \int\limits_{\Omega} T_0 \cdot \operatorname{tr}(\varepsilon(v_M)) \; dx + \int\limits_{\partial\Omega} g v_M \; dA + \int\limits_{\Omega} f v_M \; dx.$$

The thermal linear form is

$$F_T(v_T) = h \int\limits_{\partial\Omega} T_{ext} v_T \; dA.$$

By considering the global shape functions $\theta_j, \; j = 1, \ldots, N$ from the first chapter and the Basis $e_r, \; r = 1, 2, 3$ from $\mathbb{R}^3$, we define the discretized bilinear and linear forms by the following matrices and vector as :

For the purely mechanical bilinear form we obtain the following B_M matrice

$$B_{M(j,r),(l,s)} = B_M(\theta_j e_r, \theta_l e_s),$$

where $B_M \in \mathrm{Mat}(\mathbb{R}^{N\times 3})$.

Discretizing the thermal-mechanical bilinear gives

$$B_{MT(j,r),l} = B_{MT}(\theta_j e_r, \theta_l) = -\alpha(3\lambda + 2\mu) \sum_{K\in\mathfrak{T}_h} \sum_{l=1}^{l_q} \omega_{lK}\theta_l(\xi_{lK}) \operatorname{tr}(\varepsilon(\theta_j(\xi_{lK})e_r)),$$

where $B_{MT} \in \mathrm{Mat}(\mathbb{R}^{N\times 3}; \mathbb{R}^N)$.

The discretized thermal bilinear form is given by:

$$B_{Tjl} = B_T(\theta_j, \theta_l) = k \sum_{K\in\mathfrak{T}_h} \sum_{l=1}^{l_q} \omega_{lK} \nabla\theta_j(\xi_{lK}) \nabla\theta_l(\xi_{lK}) + h \sum_{F\in\mathfrak{N}_h} \sum_{l=1}^{l_q^\partial} \omega_{lF}\theta_j(\xi_{lF})\theta_l(\xi_{lF}),$$

where $B_T \in Mat(\mathbb{R}^N)$.

For the mechanical linear form we get

$$\begin{aligned} F_{M(j,f)} = F_M(\theta_j e_r) &= -\alpha(3\lambda + 2\mu) \sum_{K\in\mathfrak{T}_h} \sum_{l=1}^{l_q} \omega_{lK} T_0 \operatorname{tr}(\varepsilon(\theta_j(\xi_{lK})e_r)) \\ &\quad + \sum_{F\in\mathfrak{N}_h} \sum_{l=1}^{l_q} \omega_{lF} f(\xi_{lF})\theta_j(\xi_{lF})e_r + \sum_{K\in\mathfrak{T}_h} \sum_{l=1}^{l_q} \omega_{lK} f(\xi_{lK})\theta_j(\xi_{lK})e_r, \end{aligned}$$

where $F_M \in \mathbb{R}^{N\times 3}$.

Finally, the thermal linear form reads

$$F_{Tj} = F_T(\theta_j) = hT_{ext} \sum_{F\in\mathfrak{N}_h} \sum_{l=1}^{l_q^\partial} \omega_{lF}\theta_j(\xi_{lF})$$

where $F_T \in \mathbb{R}^N$.

After discretizing the term of the weak formulation of the partial differential equations we can write the discretized coupling thermo-mechanical system as linear system of equations

$$\underbrace{\begin{bmatrix} B_M & B_{MT} \\ 0 & B_T \end{bmatrix}}_{\mathcal{B}} \underbrace{\begin{bmatrix} U \\ T \end{bmatrix}}_{\mathcal{U}} = \underbrace{\begin{bmatrix} F_M \\ F_T \end{bmatrix}}_{\mathcal{F}}$$

The general form of our discretized linear thermo-elasticity problem is:

$$\mathcal{B}\mathcal{U} = \mathcal{F}.$$

5.4 Temperature Dependent Objective Functional

5.4.1 Larson-Miller Approach

In the last chapter, we studied the local Weibull model for Low Cycle Fatigue by considering only mechanical loads. In this chapter, we extend this approach to thermo-mechanical loads. The temperature has a local effect on the resistance of materials. Thus, we can write the number of cycles to failure as a temperature dependent variable $N_{det}(\sigma) = N_{det}(\sigma, T)$.

At elevated temperatures (most greater than half the melting point), metals exhibit creep fracture and rupture. The Larson-Miller approach can be used to determine the LCF life for materials under mechanical and thermal stress. The choose of this approach is due to the following reason: Since the migration of the dislocation lines through crystals is a diffusion process, that should be facilitated by thermally induced atomic vibrations. Therefore, the number of cycles to failure (in the range of very high temperatures) decreases with the increase of the temperature. we then define the number of cycles to failure as:

$$N_{det}(\sigma, T) = e^{-Q(T-T_0)} N_{det}(\sigma), \tag{5.3}$$

where Q stands for an activation energy characterizing the failure process and will be determined experimentally, and T_0 is another experimental constant. We can extend this Larson-Miller approach to a general temperature model with temperature dependent CMB-parameters $\sigma_f^{'}(T)$, $\varepsilon_f(T)$, $b(T)$ and $c(T)$. Combining of (3.1) and (5.3) yields

$$\varepsilon_a^{el-pl} = \frac{\sigma_f^{'}}{E} e^{-Qb(T-T_0)} (2N_{det})^b + \varepsilon_f^{'} e^{-Qc(T-T_0)} (2N_{det})^c$$

By setting $b(T) = b$, $c(T) = c$, $\sigma_f^{'}(T) = \sigma_f^{'} e^{-Qb(T-T_0)}$ and $\varepsilon_f^{'}(T) = \varepsilon_f^{'} e^{-Qc(T-T_0)}$ we obtain the temperature dependent model. For reason of simplicity, we neglect the temperature dependence of the parameters E, K and n'.

We define the objective functional as

$$\begin{aligned} J(\Omega, u, T) &= \int_{\partial\Omega} \left(\frac{1}{N_{det}(\sigma(u), T)} \right)^{\bar{m}} dA \\ &= \int_{\partial\Omega} e^{\bar{m}Q(T-T_0)} \left(\frac{1}{N_{det}(\sigma(u))} \right)^{\bar{m}} dA \end{aligned}$$

Thus, the probability of failure until load cycle t is

$$PoF(t) = 1 - e^{-t^{\bar{m}} J(\Omega, u, T)}.$$

5.4.2 Discretization of Temperature Dependent Model

The discretization of the temperature dependent LCF-model with the finite element method is analogue to the purely mechanical case described in the last chapter. We consider the global node coordinates of the mesh $X \in \mathbb{R}^{N\times 3}$ and the global degrees of freedom of the coupled systems $\mathcal{U} = \binom{U}{T} \in \mathbb{R}^{N\times 3} \times \mathbb{R}^N$. Discretizing the objective functional yields

$$\begin{aligned} J_{sur}(\Omega, u) &= \sum_{F \in \mathcal{T}_h} \int_{\widehat{F}} e^{\bar{m}Q(T(T_F(\widehat{x}))-T_0)} \left(\frac{1}{N_{det}(\sigma(T_F(\widehat{x})))} \right)^{\bar{m}} \sqrt{\det(g_F(\widehat{x}))} d\widehat{A} \\ &= \sum_{F \in \mathcal{T}_h} \sum_{l=1}^{l_q^F} \widehat{\omega}_l{}^F e^{\bar{m}Q(T(T_F(\widehat{\xi_l^F}))-T_0)} \left(\frac{1}{N_{det}(\sigma(T_F(\widehat{\xi_l^F})))} \right)^{\bar{m}} \sqrt{\det(g_F(\widehat{\xi_l^F}))}, \end{aligned}$$

where

$$T(x) = \sum_{j=1}^{N} T_j \theta_j(x)$$

is the finite element approximation of the temperature field and thus

$$T(T_F(\widehat{\xi_l^F})) = \sum_{j=1}^{N} T_{\hat{j}(K(F),j)} \hat{\theta}_j(\widehat{\xi_l^F}),$$

where $\hat{j}$ is a mapping connectivity defined in (4.9).

5.5 Sensitivity Analysis

5.5.1 Lagrangian Approach of Coupled System

In this subsection we follow the same approach described in the mechanical case and we consider the following state equation

$$\mathcal{B}(X)\mathcal{U}(X) = \mathcal{F}(X).$$

We define the Lagrange functional

$$\mathcal{L} : \mathbb{R}^{N \times 3} \times \mathbb{R}^{4N} \times \mathbb{R}^{4N} \to \mathbb{R}$$

as

$$\mathcal{L}(X,\mathcal{U},\Lambda) = J(X,\mathcal{U}) - \Lambda^T(\mathcal{B}(X)\mathcal{U}(X) - \mathcal{F}(X)),$$

where $\Lambda^T = \binom{\Lambda_M^T}{\Lambda_T^T} \in \mathbb{R}^{(N\times 3)} \times \mathbb{R}^3$ is a Lagrangian multiplier.
Derivation of the Lagrangian functional with respect to Λ yields the following state equation

$$\frac{\partial \mathcal{L}(X,\mathcal{U},\Lambda)}{\partial \Lambda} = \mathcal{B}(X)\mathcal{U}(X) - \mathcal{F}(X) = 0.$$

By deriving $\mathcal{L}(X,\mathcal{U},\Lambda)$ with respect to $\mathcal{U}$, we obtain the adjoint equation for the whole coupled system

$$\frac{\partial \mathcal{L}(X,\mathcal{U},\Lambda)}{\partial \mathcal{U}} = \frac{\partial J(X,\mathcal{U})}{\partial \mathcal{U}} - \Lambda^T\mathcal{B}(X) = 0. \tag{5.4}$$

The matrix $\mathcal{B}(X)$ in the adjoint equation (5.4) is not symmetric. Therefore, the use of standard FEM-Software is simplified by dividing the adjoint equation to a mechanical and thermal form. By splitting the adjoint equation into a system of equations we have

$$\begin{cases} B_M\Lambda_M = \left(\frac{\partial J(X,U,T)}{\partial U}\right)^T \\ B_T\Lambda_T = -B_{MT}^T\Lambda_M + \left(\frac{\partial J(X,U,T)}{\partial T}\right)^T \end{cases} \tag{5.5}$$

Note that the both matrices B_M and B_T are symmetric. The mechanical adjoint equation is coupled in the thermal adjoint equation.
Derivation of the Lagrangian functional with respect to X yields:

$$\frac{dJ(X,\mathcal{U})}{dX} = \frac{\partial \mathcal{L}(X,\mathcal{U},\Lambda)}{\partial X} = \frac{\partial J(X,\mathcal{U})}{\partial X} - \Lambda^T\Big(\frac{\partial \mathcal{B}(X)}{\partial X}\mathcal{U} - \frac{\partial \mathcal{F}(X)}{\partial X}\Big).$$

It follows

$$\frac{dJ(X,\mathcal{U})}{dX} = \frac{\partial J(X,\mathcal{U})}{\partial X} - \Lambda^T\left(\begin{bmatrix} \frac{\partial B_M(X)}{\partial X} & \frac{\partial B_{MT}(X)}{\partial X} \\ 0 & \frac{\partial B_T(X)}{\partial X}\end{bmatrix}\begin{bmatrix} U \\ T\end{bmatrix} - \begin{bmatrix} \frac{\partial F_M(X)}{\partial X} \\ \frac{\partial F_T(X)}{\partial X}\end{bmatrix}\right).$$

The total sensitivity for the coupled system is

$$\begin{aligned}\frac{dJ(X,U,T)}{dX} = \frac{\partial J(X,U,T)}{\partial X} - \Big(\Lambda_M^T \frac{\partial B_M(X)}{\partial X} U + \Lambda_M^T \frac{\partial B_{MT}(X)}{\partial X} T + \Lambda_T^T \frac{\partial B_T(X)}{\partial X} T \\ - \Lambda_M^T \frac{\partial F_M(X)}{\partial X} - \Lambda_T^T \frac{\partial F_T(X)}{\partial X}\Big). \qquad (5.6)\end{aligned}$$

A computation of the total sensitivity of the coupled thermo-mechanical system needs an extension of the code implemented in the purely mechanical case.

5.5.2 Computation of Partial Derivatives

The partial derivative $\frac{\partial B_M(X)}{\partial X}$ is already in the last chapter computed. For $\frac{\partial B_{MT}(X)}{\partial X}$ we have

$$\begin{aligned}\frac{\partial B_{MT}(e_r\theta_j,\theta_l)}{\partial X} = -\alpha(3\lambda+2\mu) \sum_{K\in\mathfrak{T}_h} \sum_{l=1}^{l_q} \Big(\overbrace{\frac{\partial}{\partial X}\big(\omega_{lK}\big)}^{already\ computed} \ \theta_l(\xi_{lK}) \cdot \operatorname{tr}\left(\varepsilon(\theta_j(\xi_{lK})e_r)\right) + \\ \omega_{lK} \underbrace{\frac{\partial}{\partial X}\big(\theta_l(\xi_{lK})\big)}_{=0} \cdot \operatorname{tr}\left(\varepsilon(\theta_j(\xi_{lK})e_r)\right) + \omega_{lK}\theta_l(\xi_{lK}) \cdot \underbrace{\frac{\partial}{\partial X}\big(\operatorname{tr}\left(\varepsilon(\theta_j(\xi_{lK})e_r)\right)\big)}_{already\ computed}\Big),\end{aligned}$$

The partial derivative of B_T with respect to X_{ji} is:

$$\begin{aligned}\frac{\partial B_T(\theta_q,\theta_k)}{\partial X_{ji}} = k \sum_{K\in\mathfrak{T}_h} \sum_{l=1}^{l_q} \Big(\overbrace{\frac{\partial}{\partial X_{ji}}\Big(\omega_{lK}\Big)}^{already\ computed} \ \nabla\theta_q(\xi_{lK}) \cdot \nabla\theta_k(\xi_{lK}) + \omega_{lK} \overbrace{\frac{\partial}{\partial X_{ji}}\Big(\nabla\theta_q(\xi_{lK})\Big)}^{to\ be\ computed} \cdot \nabla\theta_k(\xi_{lK}) \\ + \omega_{lK}\nabla\theta_q(\xi_{lK}) \cdot \underbrace{\frac{\partial}{\partial X_{ji}}\Big(\nabla\theta_k(\xi_{lK})\Big)}_{to\ be\ computed}\Big) + h \sum_{F\in\mathfrak{N}_h} \sum_{l=1}^{l_q^\partial} \Big(\underbrace{\frac{\partial}{\partial X_{ji}}\Big(\omega_{lF}\Big)}_{already\ computed} \ \theta_q(\xi_{lF})\theta_k(\xi_{lF}) + \\ \omega_{lF} \underbrace{\frac{\partial}{\partial X_{ji}}\Big(\theta_q(\xi_{lF})\Big)}_{=0} \theta_k(\xi_{lF}) + \omega_{lF}\theta_q(\xi_{lF}) \underbrace{\frac{\partial}{\partial X_{ji}}\Big(\theta_k(\xi_{lF})\Big)}_{=0}\Big)\end{aligned}$$

We need to compute the following partial derivative:

$$\frac{\partial}{\partial X_{ji}}\Big(\nabla\theta_q(\xi_{lK})\Big) = \frac{\partial}{\partial X_{ji}}\left[\Big(J_K(\hat{\xi}_l)^T\Big)^{-1}\widehat{\nabla}\widehat{\theta}_{\hat{j}_K^{-1}(q)}(\hat{\xi}_l)\right]$$

For the partial derivative of F_T w.r.t X we have:

$$\begin{aligned}\frac{\partial F_T(\theta_r)}{\partial X_{ji}} &= hT_{ext}\sum_{F\in\mathfrak{N}_h}\sum_{l=1}^{l_q^{\partial}}\Bigg(\frac{\partial}{\partial X_{ji}}\Big(\omega_{lF}\Big)\theta_r(\xi_{lF}) + \omega_{lF}\underbrace{\frac{\partial}{\partial X_{ji}}\Big(\theta_r(\xi_{lF})\Big)}_{=0}\Bigg)\\ &= hT_{ext}\sum_{F\in\mathfrak{N}_h}\sum_{l=1}^{l_q^{\partial}}\Bigg(\underbrace{\frac{\partial}{\partial X_{ji}}\Big(\omega_{lF}\Big)}_{already\ computed}\ \theta_r(\xi_{lF})\Bigg)\end{aligned}$$

For the mechanical right hand side we have

$$\begin{aligned}\frac{\partial F_M(\theta_j e_r)}{\partial X_{ji}} &= -\alpha(3\lambda+2\mu)\sum_{K\in\mathfrak{T}_h}\sum_{l=1}^{l_q}\omega_{lK}T_0\cdot\frac{\partial}{\partial X_{ji}}\Big(\operatorname{tr}\left(\varepsilon(\theta_j(\xi_{lK})e_r)\right)\Big)\\ &\quad+\frac{\partial}{\partial X_{ji}}\Big(\sum_{F\in\mathfrak{N}_h}\sum_{l=1}^{l_q}\omega_{lF}f(\xi_{lF})\theta_j(\xi_{lF})e_r\Big)+\frac{\partial}{\partial X_{ji}}\Big(\sum_{K\in\mathfrak{T}_h}\sum_{l=1}^{l_q}\omega_{lK}f(\xi_{lK})\theta_j(\xi_{lK})e_r\Big),\end{aligned}$$

and all three terms are already calculated.

For the objective functional of the coupled system we have

$$\frac{\partial J_{sur}(\Omega,u)}{\partial T_k} = \sum_{F\in\mathfrak{T}_h}\sum_{l=1}^{l_q^F}\widehat{\omega_{lF}}\frac{\partial}{\partial T_k}\Big(e^{\bar{m}Q(T(T_F(\widehat{\xi_l^F}))-T_0)}\Big)\left(\frac{1}{N_{det}(\sigma(T_F(\widehat{\xi_l^F})))}\right)^{\bar{m}},$$

with

$$\begin{aligned}\frac{\partial}{\partial T_k}\left(e^{\bar{m}Q(T(T_F(\widehat{\xi_l^F}))-T_0)}\right) &= \frac{\partial}{\partial T_k}\left(e^{\bar{m}Q(\sum_{j=1}^{N} T_j(\theta_j(\widehat{\xi_l^F}))-T_0)}\right)\\ &= \bar{m}Q\theta_k(\widehat{\xi_l^F})e^{\bar{m}Q(\sum_{j=1}^{N} T_j(\theta_j(\widehat{\xi_l^F}))-T_0)}.\end{aligned}$$

The partial derivative of $J_{sur}(\Omega, u)$ with respect to X_{ji} is

$$\begin{aligned}\frac{\partial J}{\partial X_{ji}} = \sum_{F\in\mathcal{N}_h}\sum_{l=1}^{l_q} e^{\bar{m}Q(T(T_F(\widehat{\xi_l^F}))-T_0)}\Bigg(&\underbrace{\frac{\partial}{\partial X_{ji}}\left(\omega_{lF}\right)}_{already\ computed}\left(\frac{1}{N_{det}(\sigma(T_F(\widehat{\xi_l^F})))}\right)^{\bar{m}} +\\ &\omega_{lF}\underbrace{\frac{\partial}{\partial X_{ji}}\left(\frac{1}{N_{det}(\sigma(T_F(\widehat{\xi_l^F})))}\right)^{\bar{m}}}_{already\ computed}\Bigg).\end{aligned}$$

5.6 Thermal Shape Sensitivity Computation

This section describes the steps to the computation of the thermal shape sensitivity as illustrated in Figure 5-1. Our input model X is a mechanical component subjected to mechanical and thermal stresses. After meshing the geometry, we define the material properties, boundary conditions and the applied mechanical and thermal loads. Then, we perform a steady state thermal analysis with ABAQUS to solve the heat equation in order to determine the temperature field T. The next step consists of integrating the resolved nodal temperatures in a structural analysis as predefined field and performing a coupled thermal stress analysis to get the displacement field.

We use our self implemented FEM-tool to compute the thermal cost functional and its partial derivatives with respect to displacement, shape geometry and temperature. Note that this FEM-tool contains a set of scripts implemented in R version 3.1.0, which allows us to create, modify and submit ABAQUS analysis jobs.

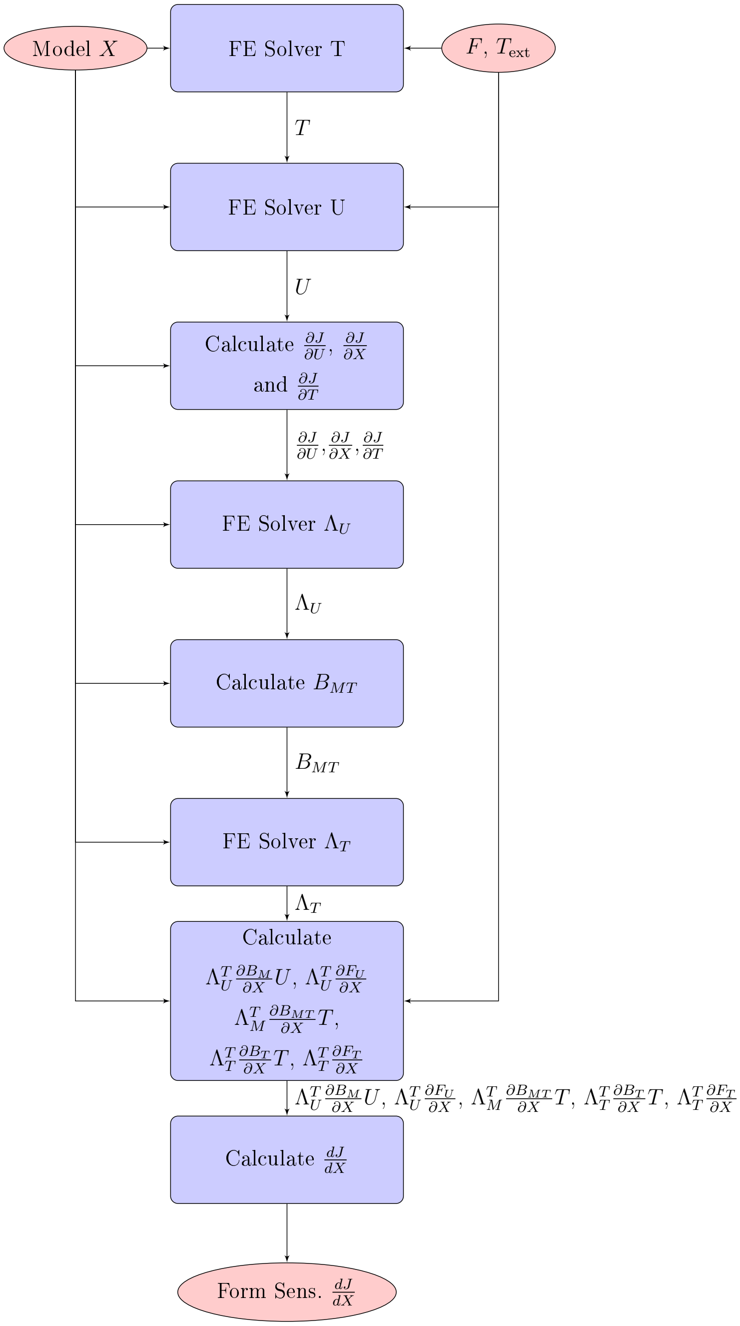


Figure 5-1: Flow diagram of shape derivative computations in thermo-mechanical case.

Furthermore we use this interface to communicate with ABAQUS by reading and visualizing the result output files. For the computation of the mechanical adjoint state Λ_U we use the partial derivative $\frac{\partial J(X,U,T)}{\partial U}$ in the right hand side of the adjoint equation (5.5) as pseudo force in the ABAQUS solver input file. Then we perform a structural analysis to get the adjoint state. We compute the thermal-mechanical matrix B_{MT} and we perform an ABAQUS run to get the thermal adjoint state Λ_T. We use our code to compute the local partial derivatives of stiffness matrices as well as the local derivatives of the force. Then, we can compute the global thermal shape sensitivity after assembling the local derivatives.

As we have seen, the computation of the total thermal sensitivity needs two runs of ABAQUS solver for the computation of the temperature and displacement fields and two runs for the computation of the mechanical and thermal adjoint state. The calculations of the the derivatives of the stiffness matrices is the most time and storage consuming step. A parallel computing approach based on the multi-core architecture is used to reduce the computational costs.

5.7 Numerical Examples and Validations

In this section, we outline the results of computations of shape sensitivity for linear thermoelasticity applied on various 2D and 3D models. We perform a sequentially coupled thermal stress analysis. In this approach, we first solve the pure heat transfer problem to obtain the nodal temperatures. Then we keep the temperatures as predefined field in a pure stress analysis to obtain the displacement field.

5.7.1 Mesh Convergence Study

As already described in the last chapter, we start by refining successively the mesh of the 2D and 3D bended rods with denser element distribution and we compute the thermal probabilistic cost functional and the Weibull scale η for each mesh refinement step.

It has to be taken into account that for every refinement step, we have to perform a steady-state thermal analysis followed by stress analysis.

Mesh density	Elements	Coordinates	Faces	Probabilistic functional	Weibull scale
Mesh 1	265	912	116	5.03e-06	445.63
Mesh 2	1060	3413	232	3.70e-05	164.32
Mesh 3	2114	6673	330	5.21e-05	138.52
Mesh 4	4220	13123	462	6.68e-05	122.32
Mesh 5	7228	22293	608	7.62e-05	114.53
Mesh 6	13572	41543	826	8.60e-05	107.83
Mesh 7	26100	79445	1144	9.38e-05	103.23
Mesh 8	47101	142844	1540	9.93e-05	100.36
Mesh 9	53534	162253	1650	1.00e-04	99.88
Mesh 10	62524	189351	1778	1.01e-04	99.27

Table 5.1: Mesh convergence for the thermal probabilistic cost functional $J(\Omega, u, T)$ and the Weibulls scale η by the 2D bended rod.

Figure 5-2 shows the converging trend of the temperature dependent Weibull scale variable η as mesh density increases. A certain (declining) level of meshing noise is observed, which however is in the same range as maximum stress variations in deterministic life calculation.

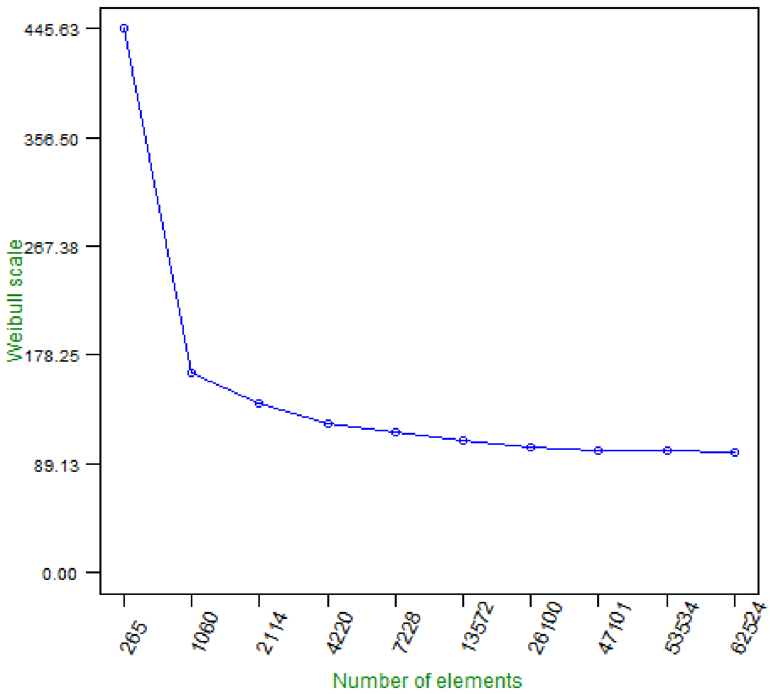


Figure 5-2: The mesh convergence for temperature dependent Weibull scales by the 2D bended rod.

Mesh density	Elements	Coordinates	Faces	Probabilistic functional	Weibull scale η
Mesh 1	570	2950	398	7.40e-11	116217.23
Mesh 2	918	4550	530	6.39e-11	125030.96
Mesh 3	1302	6410	734	8.45e-11	108789.04
Mesh 4	7700	34750	2510	1.41e-10	84182.04
Mesh 5	12816	56448	3314	1.40e-10	84254.65
Mesh 6	18824	82328	4522	1.75e-10	75561.61
Mesh 7	25520	110673	5544	1.85e-10	73501.32
Mesh 8	31330	135712	6722	1.90e-10	72449.01
Mesh 9	42582	182567	7936	2.19e-10	67539.55
Mesh 10	56610	241226	9614	2.37e-10	64892.89

Table 5.2: Mesh convergence for the thermal probabilistic cost functional $J(\Omega, u, T)$ and the Weibulls cale η by the 3D bended rod.

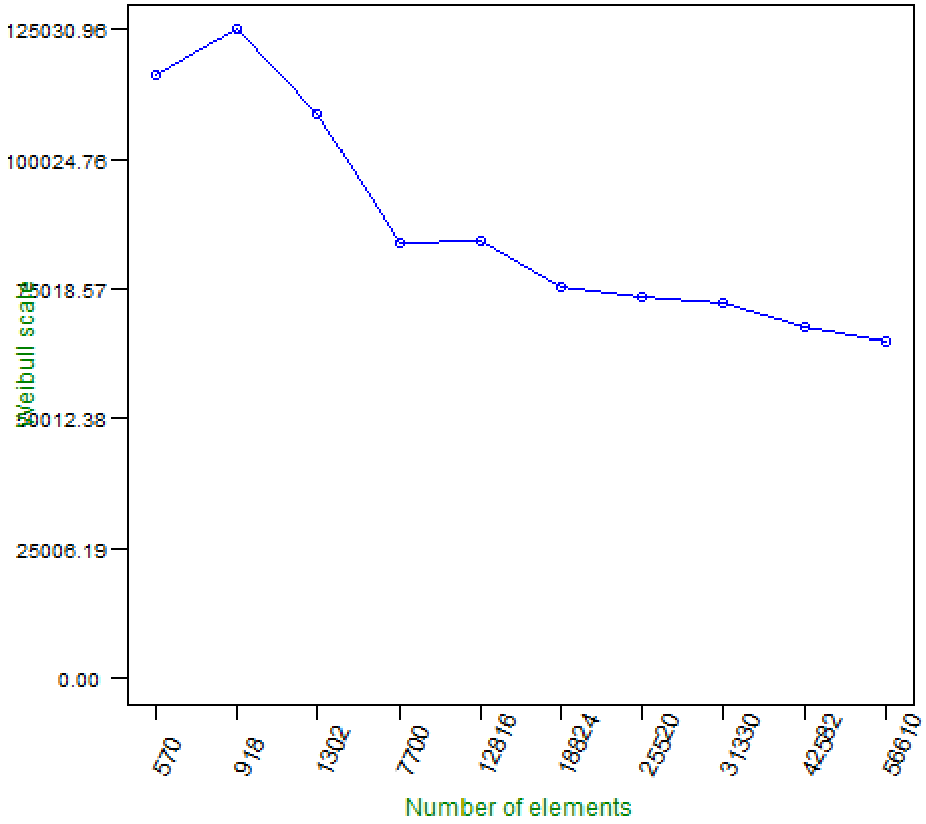


Figure 5-3: The mesh convergence for Weibull scales by the 3D bended rod.

5.7.2 Finite Element Model

In the following we describe the finite element models used in our study:

2D Model

The 2D finite element model consists out of 22293 nodes and 7228 elements. A sequential thermal-stress analysis is performed with two-dimensional, 8-node heat transfer elements, DC2D8, used for the heat transfer analysis and the corresponding 8-node plane stress continuum elements, CPS8, used for the stress analysis. The volume quadrature contains

$l_q = 9$ points and the surface quadrature contains $l_q^F = 3$ points.

3D Model

The 3D finite element model consists out of 6410 nodes and 1302 elements. We perform a sequential thermal-stress analysis with three-dimensional, 20-node heat transfer elements, DC3D20, used for the heat transfer analysis and the corresponding 20-node brick continuum elements, C3D20R, used for the stress analysis. The reduced volume quadrature contains $l_q = 8$ points and the surface quadrature contains $l_q^F = 9$ points.

5.7.3 Conductive and Convective Heat Transfer

Heat transfer between two bodies are classified into three modes: conduction, convection and radiation.

- The thermal conduction mode can be understood as the direct transfer of heat through the matter, caused by temperature difference between two adjacent regions of the same medium, or between two media in contact. Conduction does not require any bulk motion of matter. This type of heat transfer can occur through a gas turbine blade.

- The thermal convection mode is caused by the transfer of heat energy between a surface and a moving fluid. This type of transfer takes place in gas turbine industry by cooling of turbine blades with air, for example.

- The thermal radiation mode is caused by the transfer of energy through space without the necessary presence of matter.

In this subsection we focus on the thermal shape sensitivity by conductive and convective heat transfer. We consider two different models: A 2D and 3D bended rod subjected to thermal stress and tensile loading.

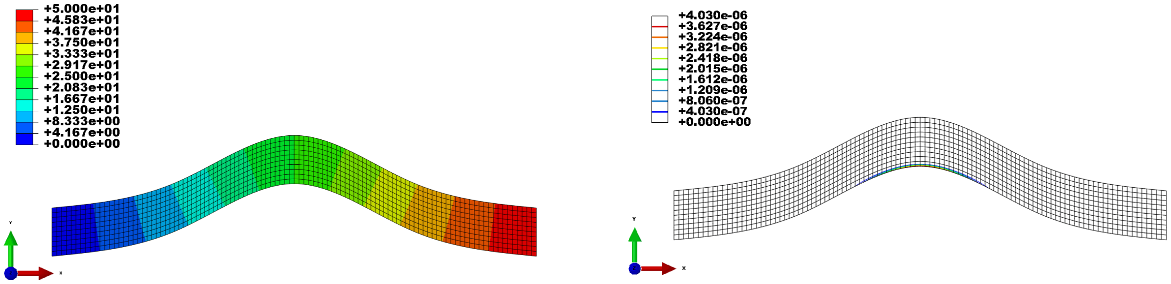
In case of conductive heat transfer, the clamped sides of the 2D rod and the 3D rod are exposed to a low temperature of $273,15\ K$ and the right sides are subjected to a high temperature of $323,15\ K$.

In case of convective heat transfer, the surfaces on the upper and lower sides of the 2D bended rod are exposed to a sink temperature $T_{s_1} = 373,15\ K$ with convective heat transfer coefficient $h_1 = 20\ W/m^2K$. The surfaces on the left and the right sides are exposed to a sink temperature $T_{s_2} = 273,15\ K$ with convective heat transfer coefficient $h_1 = 20\ W/m^2K$. The two surfaces on the left and the right sides of the 3D bended rod are subjected to a sink temperature $T_{s_1} = 323,15\ K$ with convective heat coefficient $h_2 = 20\ W/m^2K$. The surrounding surface is subjected to a sink Temperature $T_{s_2} = 273,15\ K$ with convective heat coefficient $h_2 = 20\ W/m^2K$.

The thermal material parameters used in both cases are the thermal conductivity $k = 160\ W/mK$, the thermal expansion $\alpha = 2.3 \times 10^{-5}$ and the activation energy constant $Q = -\log(1/2)/100$. The surfaces on the right side of the 2D rod and the 3D rod are subjected to an uniform traction loading with an amplitude of $75\ N/mm^2$ and $12\ N/mm^2$ respectively.

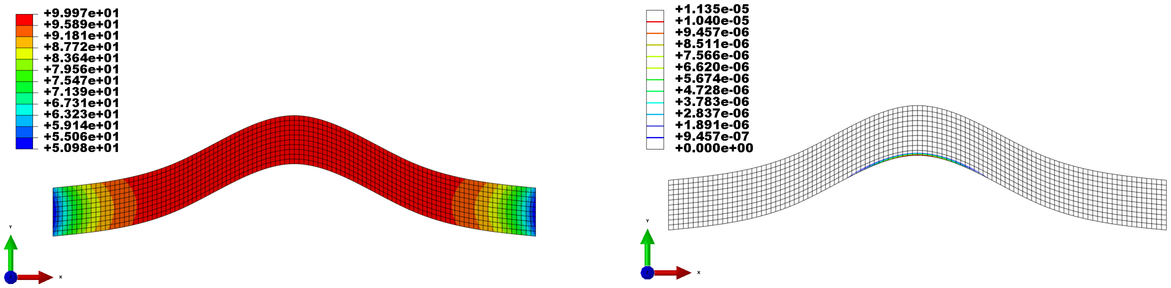
Before we begin with the computation of the thermal cost functional J_{sur}, we have to solve the heat equation in order to obtain the nodal temperatures. Figures 5-4a, 5-4c, 5-5a and 5-5c show the temperature distribution along the rods after solving the heat equation by conductive and convective heat transfer. Figures 5-4b, 5-4d, 5-5b and 5-5d show the crack formation intensity by linear thermoelasticity for the 2D and 3D bended rods. The probability of failure increases on the lower side of the location with the strongest curvature, where the thermal stress concentration is maximal both in conductive and convective cases.

The three partial derivatives $\frac{\partial J}{\partial U}$, $\frac{\partial J}{\partial X}$ and $\frac{\partial J}{\partial T}$ can be obtained in direction Z by the following



(a) The temperature distribution along the 2D bended rod by conductive heat transfer.

(b) Thermal crack formation intensity for the 2D bended rod by conductive heat transfer.



(c) The temperature distribution along rod by convective heat transfer.

(d) Thermal crack formation intensity for the rod by convective heat transfer.

Figure 5-4: The temperature distribution and thermal crack formation intensity by the 2D rod J.

equations:

$$\begin{aligned} Z_i^u.\frac{\partial J}{\partial U} &= \lim_{\epsilon \to 0} \frac{J(X, U + \epsilon Z_i^u, T) - J(X, U, T)}{\epsilon}, \\ Z_i^x.\frac{\partial J}{\partial X} &= \lim_{\epsilon \to 0} \frac{J(X + \epsilon Z_i^x, U, T) - J(X, U, T)}{\epsilon} \text{ and} \\ Z_i^t.\frac{\partial J}{\partial T} &= \lim_{\epsilon \to 0} \frac{J(X, U, T + \epsilon Z_i^t) - J(X, U, T)}{\epsilon} \end{aligned}$$

where Z_i^u, Z_i^x and Z_i^t are small random perturbations of the displacements, coordinates and temperatures field respectively.

We use the following finite difference approximation for the total thermal shape sensitivity:

$$Z_i^x . \frac{dJ}{dX} \approx \frac{J(X+\epsilon Z_i^x, U(X+\epsilon Z_i^x), T(X+\epsilon Z_i^x)) - J(X,U,T)}{\epsilon}$$

Note that for every random perturbation Z_i^x $(i = 1, ..., 6)$, we have to run the FE solver to solve the heat equation in order to compute the new temperature field $T(X + \epsilon Z_i^x)$ correspond to the new geometry $X + \epsilon Z_i^x$. Then, we compute the new displacement field $U(X + \epsilon Z_i^x)$ correspond to the perturbed geometry $X + \epsilon Z_i^x$.

The partial derivatives $\frac{\partial J}{\partial U}$, $\frac{\partial J}{\partial X}$ and $\frac{\partial J}{\partial T}$ by conductive and convective heat transfer for the 2D and 3D bended rod are shown in figures 5-6(a,c,e) and 5-7(a,c,e). Validation results of the computed partial derivatives divided by finite differences for various stepsizes and random directions are displayed in the figures 5-6(b,d,f) and 5-7(b,d,f). We show that the range of the relative errors is 0.1%.

The total thermal shape sensitivities for the convective and conductive heat transfer by the 2D and 3D bended rod are shown in figures 5-10 and 5-11. As one can see, the direction of improved reliability given by the negative shape gradient points downward and outward. The outward direction aims to diminish the risk of LCF failure by adding more material. The longest arrows in the downward direction in the middle part in the bended rods clearly aims to reduce stress concentration at the critical spot.

Numerical validation work has been conducted by comparison of the shape gradients with finite difference calculations for different stepsizes and random geometry perturbations. Results are reported in right panels in figures 5-10 and 5-11. The relative error is less than 0.1%.

Quantity	**Elapsed**				**Cores**	**Tool**
	2D		**3D**			
	conv	cond	conv	cond		
Temperature T	23.36	25.89	25.46	25.36	1	`ABAQUS CAE 6.13`
Displacement U	25.34	26.54	25.36	25.34	1	`ABAQUS CAE 6.13`
$\frac{\partial J}{\partial U}$,$\frac{\partial J}{\partial X}$ and $\frac{\partial J}{\partial T}$ together	2.51	3.52	20.40	20.10	1	`R 3.1.0`
Adjoint State Λ_U	25.37	26.55	25.39	25.31	1	`ABAQUS CAE 6.13`
Adjoint State Λ_T	25.34	26.62	27.91	27.32	1	`ABAQUS CAE 6.13`
$\frac{\partial B_M}{\partial X}$, $\frac{\partial B_{MT}}{\partial X}$, $\frac{\partial B_T}{\partial X}$, $\frac{\partial F_M}{\partial X}$ and $\frac{\partial F_T}{\partial X}$ together	73.18	64.63	1763.85	1704	1	`R 3.1.0`
$\frac{\partial B_M}{\partial X}$, $\frac{\partial B_{MT}}{\partial X}$, $\frac{\partial B_T}{\partial X}$, $\frac{\partial F_M}{\partial X}$ and $\frac{\partial F_T}{\partial X}$ together (parallel)	30.11	27.50	414.51	431.42	6	`R 3.1.0`
Sensitivity dJ/dX	156.89	145.31	716.34	691.20	6	`ABAQUS CAE 6.13` `R 3.1.0`

Table 5.3: Execution times in sec for thermal sensitivity by conductive (cond) and convective (conv) heat transfer on an Intel Core i7-3630QM CPU @ 2.40GHZ, 8GB shared memory machine with 4 physical and 8 virtual cores.

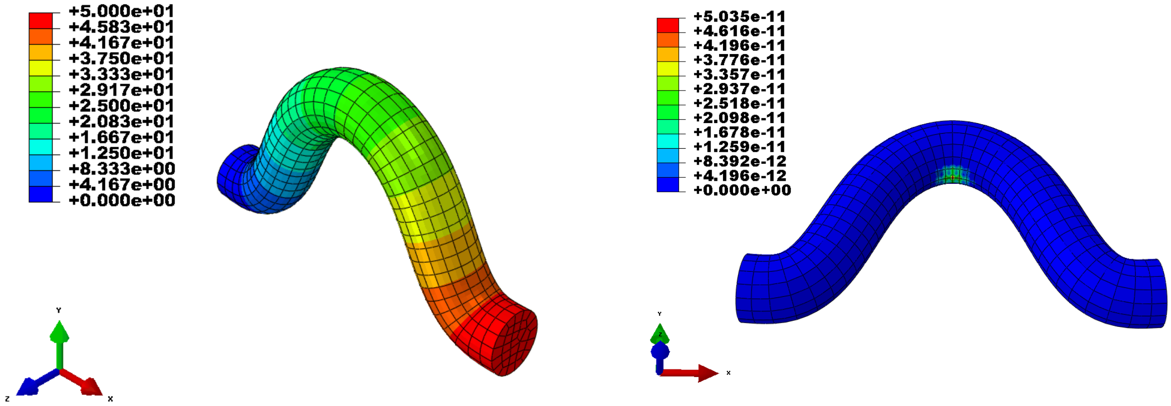
As can be seen from the table 5.3, most CPU-time to evaluate the total thermal sensitivity $\frac{dJ}{dX}$ is consumed by computing the partial derivatives of the stiffness matrices and forces (i.e. $\frac{\partial B_M}{\partial X}$, $\frac{\partial B_{MT}}{\partial X}$, $\frac{\partial B_T}{\partial X}$, $\frac{\partial F_M}{\partial X}$ and $\frac{\partial F_T}{\partial X}$). This is mainly due to the fact that even local shape derivatives of the mechanical stiffness matrix require the calculation of an array (tensor) of the dimension $3 \times n_{sh} \times 3 \times n_{sh} \times 3 \times n_{sh} \times l_q$, where n_{sh} stands for the local degree of freedom and l_q is the number of quadratures. For a brick element 3D20R with 20 points and 8 quadratures, the term $\frac{\partial B_M}{\partial X}$ requires 1.728 million array entries (see table 5.4 for other terms). Due to the support of multi-core processor technology and the use of R-package parallel, computations were carried out to significantly reduce the simulation run-times.

Quantity	**Size**	**Entries by C3D20R element**
$\frac{\partial B_M}{\partial X}$	$3 \times n_{sh} \times 3 \times n_{sh} \times 3 \times n_{sh} \times l_q$	1 728 000
$\frac{\partial B_{MT}}{\partial X}$	$3 \times n_{sh} \times n_{sh} \times 3 \times n_{sh} \times l_q$	576 000
$\frac{\partial B_T}{\partial X}$	$n_{sh} \times n_{sh} \times 3 \times n_{sh} \times l_q$	192 000
$\frac{\partial F_M}{\partial X}$	$3 \times n_{sh} \times 3 \times n_{sh} \times l_q$	28 800
$\frac{\partial F_T}{\partial X}$	$n_{sh} \times 3 \times n_{sh} \times l_q$	9600

Table 5.4: The sizes of tensors involved in the thermal shape sensitivity computations

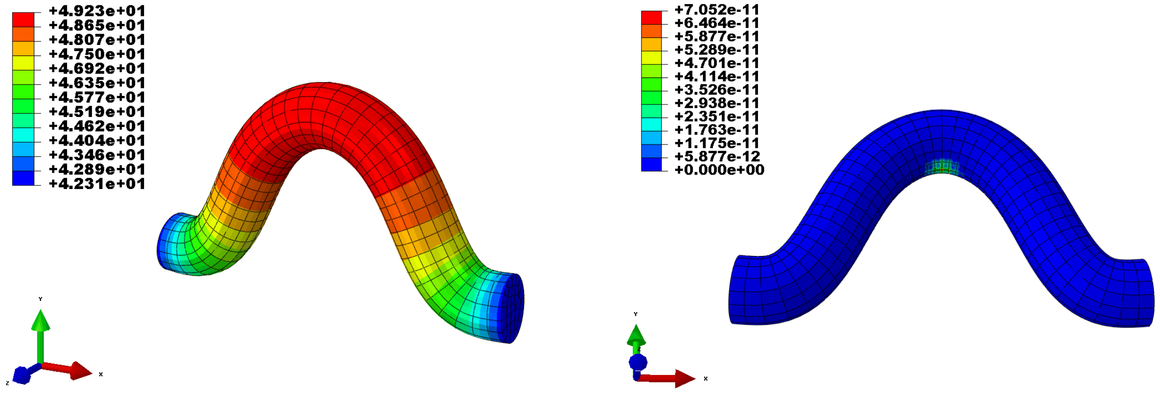
5.8 Conclusion

A new probabilistic model based on Larson-Miller approach for computing the thermal sensitivity of mechanical components has been presented. The sequentially coupled approach, where the strain effect on temperature can be neglected, has therefore been chosen instead of the fully coupled approach. Several numerical 2D and 3D models of thermo-mechanical sensitivity analysis have been presented in this chapter. The computations of thermal shape gradients using two adjoint equations have been shown to give very accurate results by comparison with finite difference calculations. Execution time efficiency has been increased using parallel computations based on multi-core architecture.



(a) The temperature distribution along the 3D rod by conductive heat transfer.

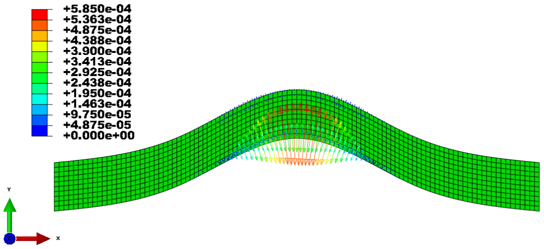
(b) Thermal crack formation intensity for the 3D rod by conductive heat transfer.



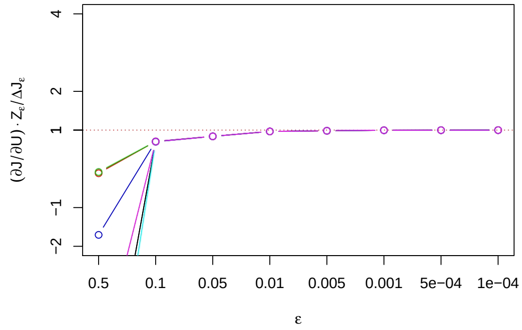
(c) The temperature distribution along the 3D rod by convective heat transfer.

(d) Thermal crack formation intensity for the 3D rod by convective heat transfer.

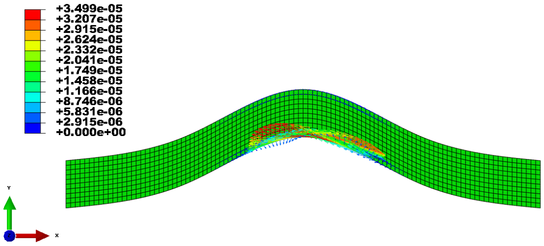
Figure 5-5: The temperature distribution and thermal crack formation intensity by the 3D rod J.



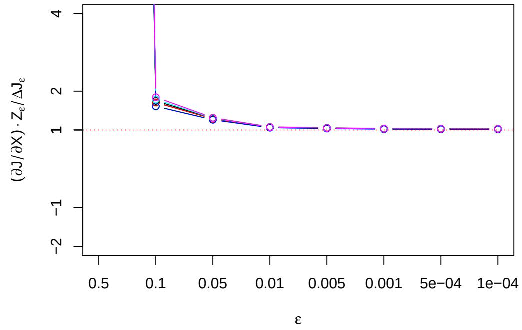
(a) Partial derivative $\frac{\partial J}{\partial U}_j$ visualized as an arrow at node X_j.



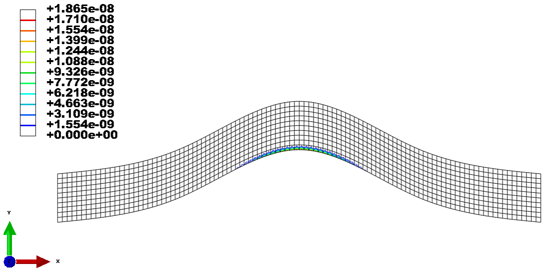
(b) Comparison of $\frac{\partial J}{\partial U} \cdot Z_i$ with the finite differences $\frac{J(X,U+\varepsilon Z_i,T)-J(X,U,T)}{\varepsilon}$ for $\varepsilon \to 0$. Z_i is a random perturbation field ($i = 6$).



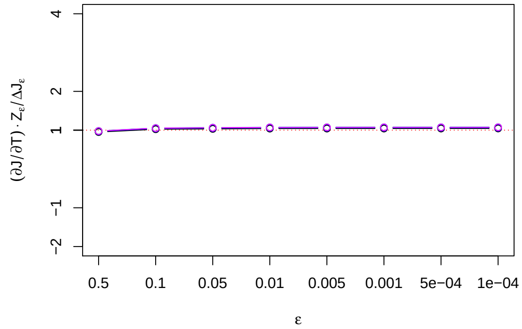
(c) The jth component of $\frac{\partial J}{\partial X}$ is visualized as an arrow at node X_j.



(d) Comparison of $\frac{\partial J}{\partial X} \cdot Z_i$ with the finite differences $\frac{J(X+\varepsilon Z_i,U,T)-J(X,U,T)}{\varepsilon}$ for $\varepsilon \to 0$. Z_i is a random perturbation field ($i = 6$).

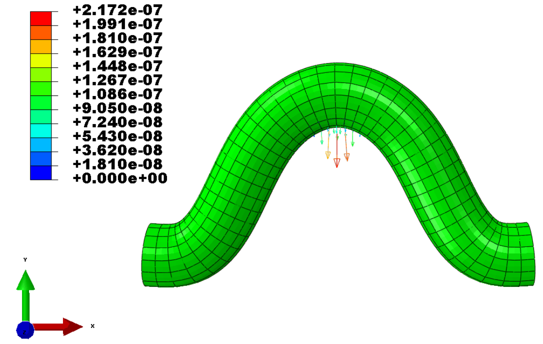


(e) Visualization of $\frac{\partial J}{\partial T}$.

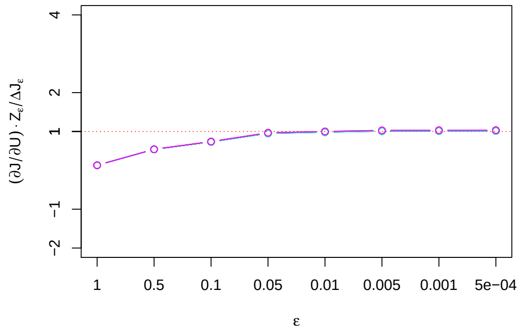


(f) Comparison of $\frac{\partial J}{\partial T} \cdot Z_i$ with the finite differences $\frac{J(X,U,T+\varepsilon Z_i)-J(X,U,T)}{\varepsilon}$ for $\varepsilon \to 0$. Z_i is a random perturbation field ($i = 6$).

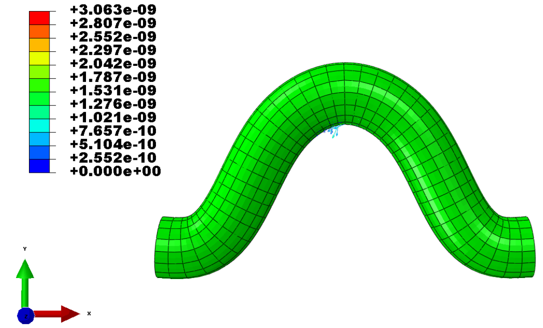
Figure 5-6: Partial derivatives of the thermal cost functional by conductive heat transfer J.



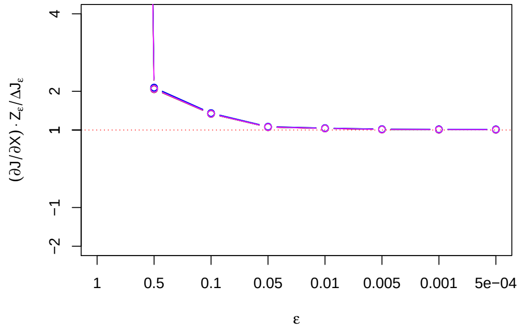
(a) Partial derivative $\frac{\partial J}{\partial U}_j$ visualized as an arrow at node X_j.



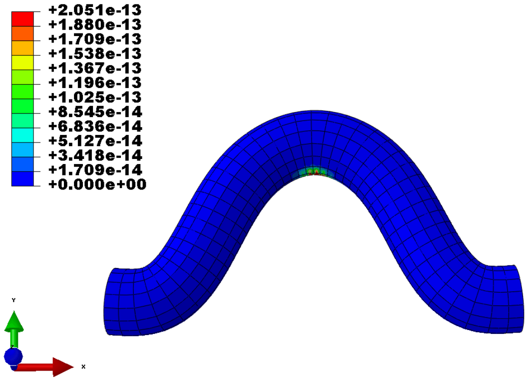
(b) Comparison of $\frac{\partial J}{\partial U} \cdot Z_i$ with the finite differences $\frac{J(X,U+\varepsilon Z_i,T)-J(X,U,T)}{\varepsilon}$ for $\varepsilon \to 0$. Z_i is a random perturbation field ($i = 6$).



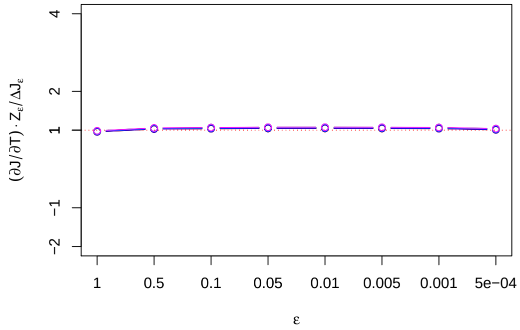
(c) The jth component of $\frac{\partial J}{\partial X}$ is visualized as an arrow at node X_j.



(d) Comparison of $\frac{\partial J}{\partial X} \cdot Z_i$ with the finite differences $\frac{J(X+\varepsilon Z_i,U,T)-J(X,U,T)}{\varepsilon}$ for $\varepsilon \to 0$. Z_i is a random perturbation field ($i = 6$).

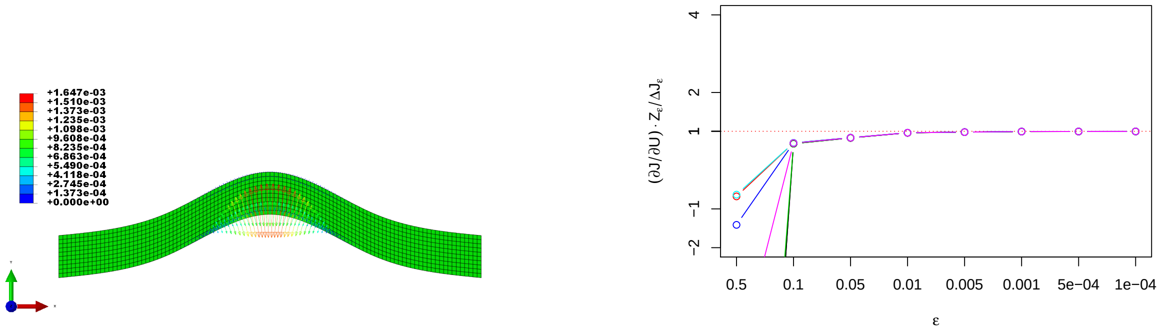


(e) Visualization of $\frac{\partial J}{\partial T}$.



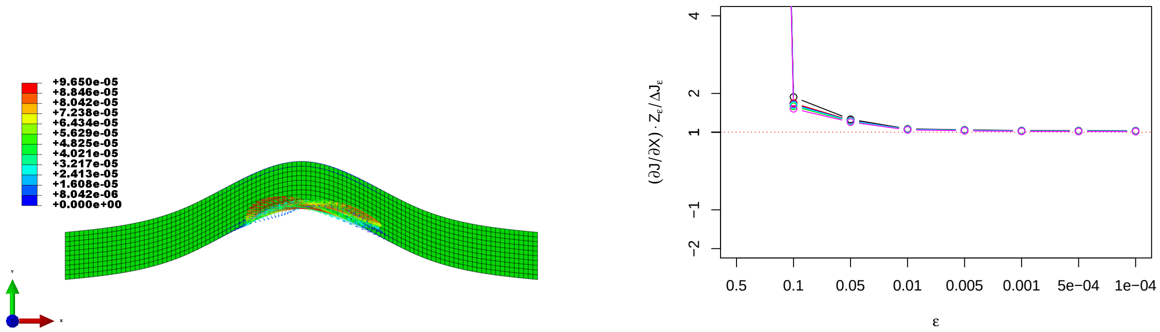
(f) Comparison of $\frac{\partial J}{\partial T} \cdot Z_i$ with the finite differences $\frac{J(X,U,T+\varepsilon Z_i)-J(X,U,T)}{\varepsilon}$ for $\varepsilon \to 0$. Z_i is a random perturbation field ($i = 6$).

Figure 5-7: Partial derivatives of the thermal cost functional J for 3D rod by conductive heat transfer.



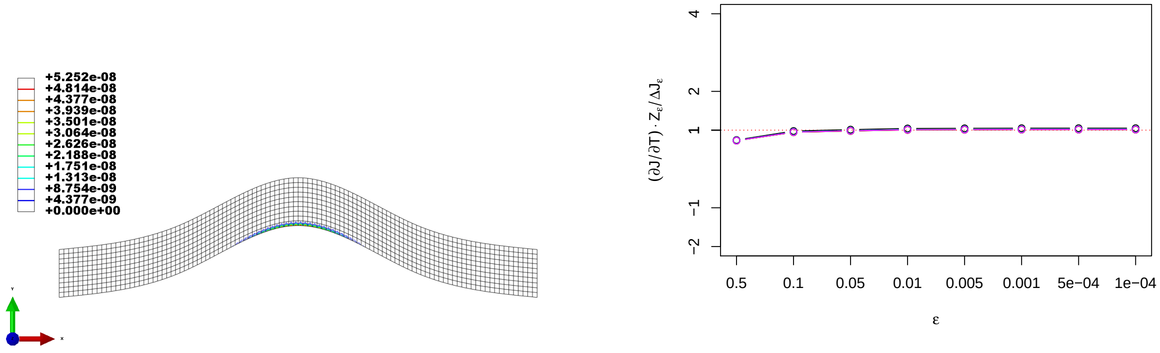
(a) Partial derivative $\frac{\partial J}{\partial U}_j$ visualized as an arrow at node X_j.

(b) Comparison of $\frac{\partial J}{\partial U} \cdot Z_i$ with the finite differences $\frac{J(X,U+\varepsilon Z_i,T)-J(X,U,T)}{\varepsilon}$ for $\varepsilon \to 0$. Z_i is a random perturbation field ($i = 6$).



(c) The jth component of $\frac{\partial J}{\partial X}$ is visualized as an arrow at node X_j.

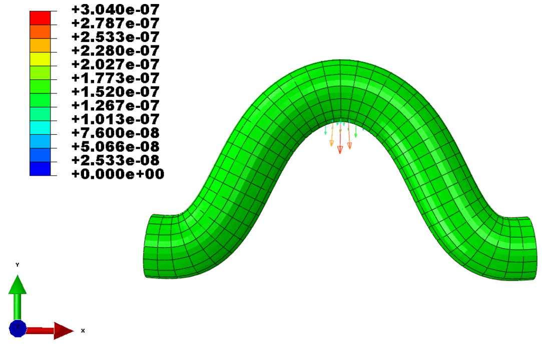
(d) Comparison of $\frac{\partial J}{\partial X} \cdot Z_i$ with the finite differences $\frac{J(X+\varepsilon Z_i,U,T)-J(X,U,T)}{\varepsilon}$ for $\varepsilon \to 0$. Z_i is a random perturbation field ($i = 6$).



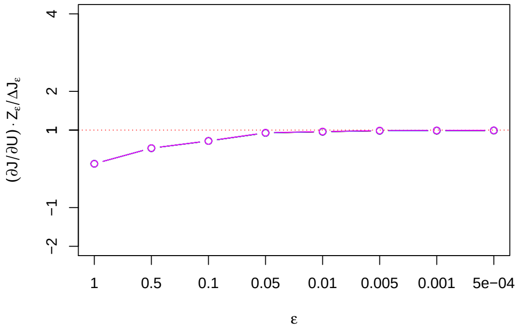
(e) Visualization of $\frac{\partial J}{\partial T}$.

(f) Comparison of $\frac{\partial J}{\partial T} \cdot Z_i$ with the finite differences $\frac{J(X,U,T+\varepsilon Z_i)-J(X,U,T)}{\varepsilon}$ for $\varepsilon \to 0$. Z_i is a random perturbation field ($i = 6$).

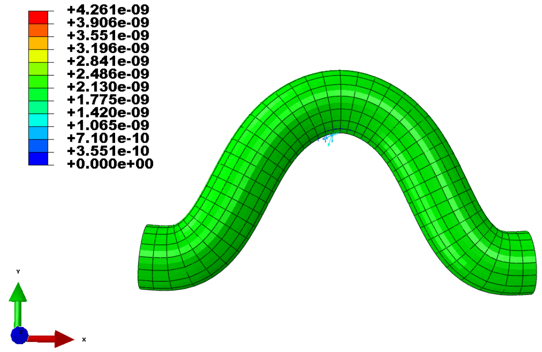
Figure 5-8: Partial derivatives of the thermal cost functional J by convective heat transfer.



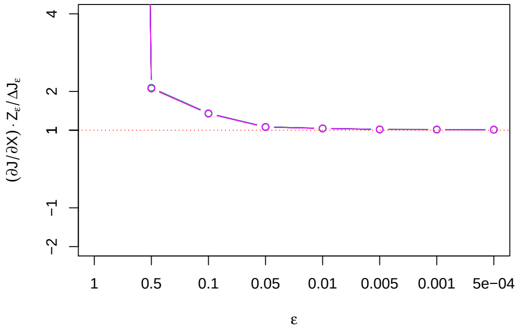
(a) Partial derivative $\frac{\partial J}{\partial U}_j$ visualized as an arrow at node X_j.



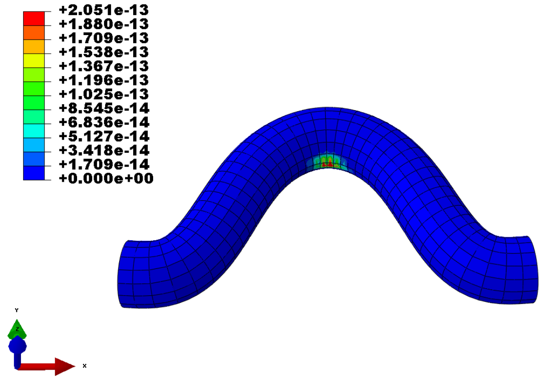
(b) Comparison of $\frac{\partial J}{\partial U} \cdot Z_i$ with the finite differences $\frac{J(X,U+\varepsilon Z_i,T)-J(X,U,T)}{\varepsilon}$ for $\varepsilon \to 0$. Z_i is a random perturbation field ($i = 6$).



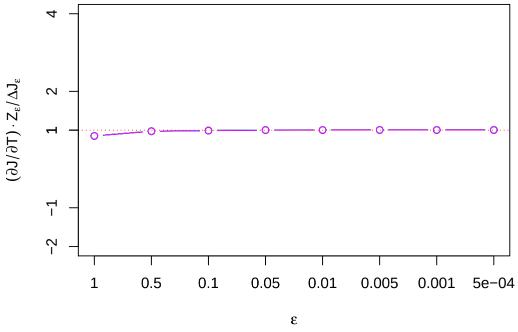
(c) The jth component of $\frac{\partial J}{\partial X}$ is visualized as an arrow at node X_j.



(d) Comparison of $\frac{\partial J}{\partial X} \cdot Z_i$ with the finite differences $\frac{J(X+\varepsilon Z_i,U,T)-J(X,U,T)}{\varepsilon}$ for $\varepsilon \to 0$. Z_i is a random perturbation field ($i = 6$).

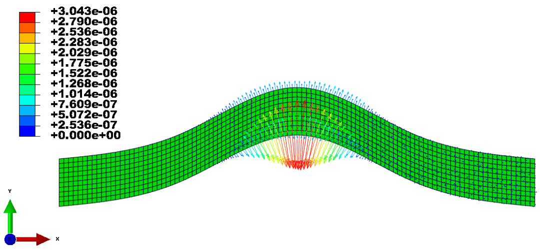


(e) Visualization of $\frac{\partial J}{\partial T}$.

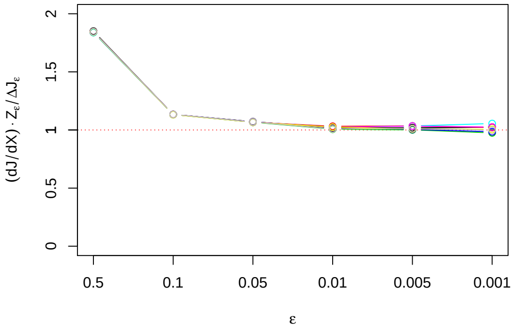


(f) Comparison of $\frac{\partial J}{\partial T} \cdot Z_i$ with the finite differences $\frac{J(X,U,T+\varepsilon Z_i)-J(X,U,T)}{\varepsilon}$ for $\varepsilon \to 0$. Z_i is a random perturbation field ($i = 6$).

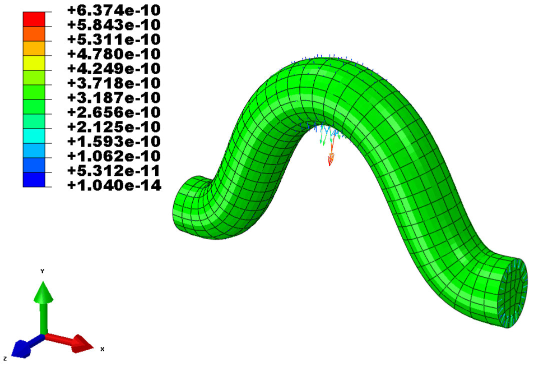
Figure 5-9: Partial derivatives of the thermal cost functional J by convective heat transfer.



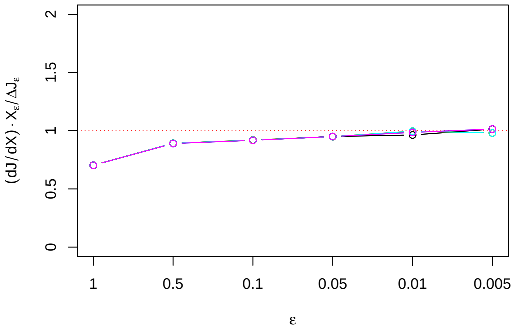
(a) The total shape sensitivity $\frac{dJ}{dX}$. The jth component of $\frac{dJ}{dX}$ is visualized as an arrow at node X_j.



(b) Comparison of the shape sensitivity $\frac{dJ}{\partial X} \cdot Z_i$ with the finite differences $\frac{J(X+\varepsilon Z_i, U(X+\varepsilon Z_i), T(X+\varepsilon Z_i)) - J(X,U,T)}{\varepsilon}$ for $\varepsilon \to 0$. Z_i is a random perturbation field.

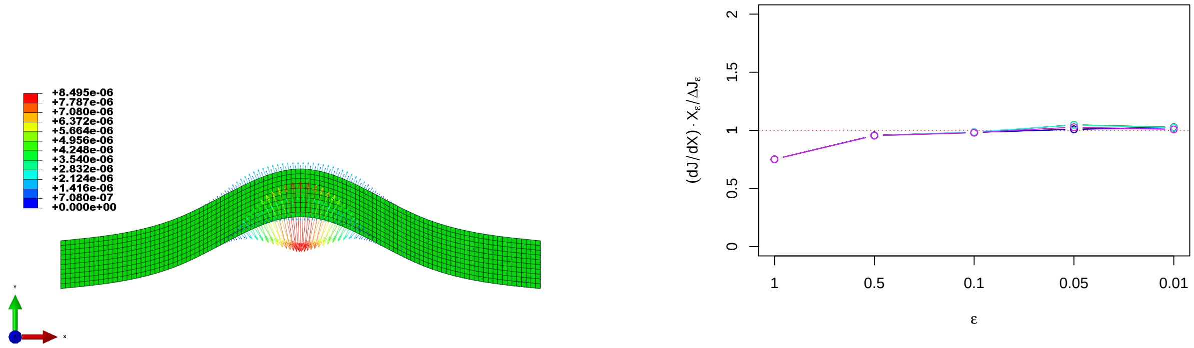


(c) The total shape sensitivity $\frac{dJ}{dX}$. The jth component of $\frac{dJ}{dX}$ is visualized as an arrow at node X_j.



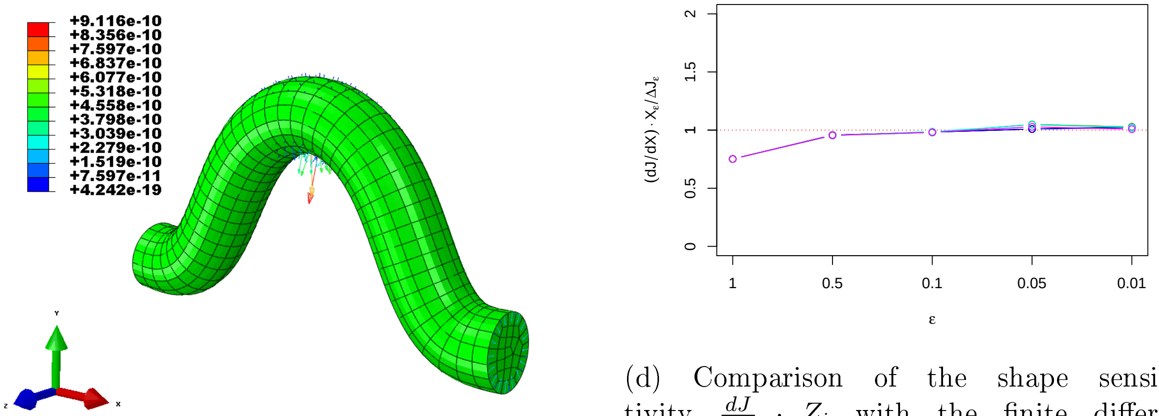
(d) Comparison of the shape sensitivity $\frac{dJ}{\partial X} \cdot Z_i$ with the finite differences $\frac{J(X+\varepsilon Z_i, U(X+\varepsilon Z_i), T(X+\varepsilon Z_i)) - J(X,U,T)}{\varepsilon}$ for $\varepsilon \to 0$. Z_i is a random perturbation field.

Figure 5-10: The thermal shape sensitivity and validation by conductive heat transfer.



(a) The total shape sensitivity $\frac{dJ}{dX}$. The jth component of $\frac{dJ}{dX}$ is visualized as an arrow at node X_j.

(b) Comparison of the shape sensitivity $\frac{dJ}{\partial X} \cdot Z_i$ with the finite differences $\frac{J(X+\varepsilon Z_i, U(X+\varepsilon Z_i), T(X+\varepsilon Z_i)) - J(X,U,T)}{\varepsilon}$ for $\varepsilon \to 0$. Z_i is a random perturbation field.



(c) The total shape sensitivity $\frac{dJ}{dX}$. The jth component of $\frac{dJ}{dX}$ is visualized as an arrow at node X_j.

(d) Comparison of the shape sensitivity $\frac{dJ}{\partial X} \cdot Z_i$ with the finite differences $\frac{J(X+\varepsilon Z_i, U(X+\varepsilon Z_i), T(X+\varepsilon Z_i)) - J(X,U,T)}{\varepsilon}$ for $\varepsilon \to 0$. Z_i is a random perturbation field.

Figure 5-11: The thermal shape sensitivity and validation by convective heat transfer.

Chapter 6

Conclusion

This thesis covers different topics in the field of structural analysis and shape sensitivity analysis concerning the development of methods and tools for the assessment of the shape sensitivity by a probabilistic model for material fatigue. Based on advanced life prediction methods, this shape gradient tool shall be able to deliver reliable information to engineers to improve the quality of design as a part of an advanced future design process.

Linear Elasticity and Finite Element Method

The mechanical behavior of metallic components under stress loading was described by partial differential equation. Furthermore, a short mathematical introduction to the finite element method concerning mesh generation, construction of finite elements, Galerkin approximation and numerical integrations was presented.

Discretization of the probabilistic LCF model

Based on the local and probabilistic model developed in [43] we used the finite element approximation to calculate the probabilistic cost functional, which is an integral over a surface containing mechanical and thermal stress informations. The finite element tool has been implemented in R and contains various shape functions for 1D, 2D and 3D isoparametric elements. Our FEM-tool has been adapted with the element library of the commercialized solver ABAQUS/Standard to ensure an effective, fast and error-free computations. Some basic conceptual examples as complex models for various 2D and 3D models has been constructed, implemented and studied to validate and visualize the crack formation intensity.

Sensitivity Analysis : Linear Elasticity

The effects of design geometry and the production-related shape perturbations are the aim of our sensitivity analysis. Two approaches for computing the shape sensitivity are presented in chapter 4. As the number of surface geometry parameters is too large, we use the adjoint method which guarantees an efficient computation of shape gradients. A detailed derivation of the adjoint equation is described in the subsection 4.1.2. All partial derivatives needed for the computation of the total shape sensitivity are computed and implemented in section 4.1. To demonstrate the capabilities of our tool several test cases ranging from simple 2D examples to complex industrial ones are performed. The partial derivatives of the probabilistic cost functional are benchmarked and validated against finite-difference approximations by a small perturbation of the shape geometry. It has been shown that there is an excellent agreement between the computed adjoint shape gradient and the finite difference derivative approximation.

Sensitivity Analysis : Linear Thermoelasticity

Our finite element tool has been developed to take in account not only the mechanical loading but also the thermal stress caused by the change of temperatures in the state of design components. An algebraic thermo-mechanical system of equations coupling the heat transfer equation and the governing linear thermoelasticity equation was derived. The temperature dependent objective functional based on Larson-Miller approach described in section 5.4 has been discretized using isoparametric elements for displacement, geometry and temperature. An efficient Lagrangian approach using two adjoint equations has been presented to compute the thermal shape gradients. The feasibility of our tool and the quality of the sensitivity gradients are judged by comparing our results with the finite difference methods.

It was observed that the evaluation of the probabilistic shape gradient with the adjoint approach offers an efficient method for designer to analyse the effect of geometry on the life time of design components. Our framework has been tested by 2D and 3D models with different stress and thermal loadings and has been verified against finite-difference derivative approximations with relative errors in the range of one percent. According to the best of our knowledge, this is the first numerical evaluation of shape gradients based on failure probabilities in the thermomechanical setting.

A worthwhile continuation of the presented work would be to integrate our tool in other numerical shape optimization framework to decrease the failure probability of ceramic structures by three dimensional geometries. A two dimensional approach is described in [8]. Moreover, our tool shall be used widely to different research and development fields. It is necessary to take into account some engineering constraints like contact boundary conditions (see [28]). Another future approach could be the extension of the probabilistic LCF functional to further material classes and damage mechanisms, especially considering notch effects [36].

Appendix A

Steady-state heat equation

For the case of steady-state (time-independent or stationary state) and source-free temperature field , the heat equation will described by the Laplace equation

$$\Delta T = 0.$$

By considering the Dirichlet conditions $T = T_0$ on the boundaries $\partial\Omega$, we can write

$$\begin{cases} \Delta T = 0, & \text{in } \Omega; \\ T = T_0, & \text{on } \partial\Omega. \end{cases} \tag{A.1}$$

When the temperature T_0 is not necessarily equal to zero, we read about a partial differential equation (A.1) with non-homogenous Dirichlet conditions.
We set $T^* = T - T_0$ and we consider the test space $V_D = \{v \in H^1(\Omega),\ v = 0 \text{ on } \partial\Omega\}$. Thus,

$$\begin{cases} \Delta T^* = -\Delta T_0, & \text{in } \Omega; \\ T^* = 0, & \text{on } \partial\Omega. \end{cases} \tag{A.2}$$

An integration by parts using Green's identity yields:

$$\int_{\Omega} \Delta T^* v \; dx = -\int_{\Omega} \nabla T^* \cdot \nabla v \; dx + \int_{\partial\Omega} (\nabla T^* \cdot n) v \; dA = -\int_{\Omega} \nabla T^* \cdot \nabla v \; dx.$$

The same yields for T_0

$$\int_{\Omega} \Delta T_0 v \; dx = -\int_{\Omega} \nabla T_0 \cdot \nabla v \; dx.$$

The weak formulation of the Poisson equation shows:

$$\begin{cases} \text{Find } T^* \in V_D \text{ so that} \\ a(T^*, v) = F(v), \end{cases} \tag{A.3}$$

where $a(T^*, v) = \int_{\Omega} \nabla T^* \cdot \nabla v \; dx$ and $F(v) = -\int_{\Omega} \nabla T_0 \cdot \nabla v \; dx$.
The thermal bilinear form B_{T^*} and the linear form F_T will be have the forms:

$$B_{T^*}(T^*, v) = \int_{\Omega} \nabla T^* \cdot \nabla v \; dx \text{ and } F_T(v) = -\int_{\Omega} \nabla T_0 \cdot \nabla v \; dx,$$

where $v \in H_0^1(\Omega)$.

Appendix B

Shape Functions

B.1 Shape Functions for 1D Elements

B.1.1 1D Linear Element

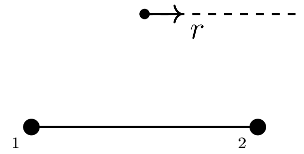


Figure B-1: Linear 1D element.

The shape functions of the linear 1D element and its derivatives with respect to the local coordinate r are:

$$\phi_1(r) = 1/2(1-r); \qquad \frac{\partial \phi_1}{\partial r}(r) = -1/2;$$

$$\phi_2(r) = 1/2(1+r); \qquad \frac{\partial \phi_1}{\partial r}(r) = 1/2;$$

B.1.2 1D Quadratic Element

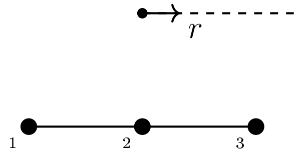


Figure B-2: Qudratic 1D element.

The shape functions for the 1D quadratic element and its The derivatives with respect to the local coordinate r are :

$$\phi_1(r) = r(r-1)/2; \qquad \frac{\partial \phi_1}{\partial r} = r - 1/2;$$
$$\phi_2(r) = (1-r)(1+r); \qquad \frac{\partial \phi_2}{\partial r} = -2r;$$
$$\phi_3(r) = r(1+r)/2; \qquad \frac{\partial \phi_3}{\partial r} = r + 1/2.$$

B.2 Shape Functions for 2D Elements

B.2.1 3-Node 2D Linear Triangular Element

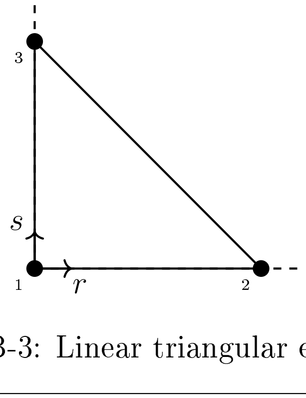


Figure B-3: Linear triangular element.

$$\begin{array}{lll}
\phi_1(r,s) = 1 - r - s; & \dfrac{\partial \phi_1}{\partial r}(r,s) = -1; & \dfrac{\partial \phi_1}{\partial s}(r,s) = -1; \\
\phi_2(r,s) = r; & \dfrac{\partial \phi_2}{\partial r}(r,s) = 1; & \dfrac{\partial \phi_2}{\partial s}(r,s) = 0; \\
\phi_3(r,s) = s; & \dfrac{\partial \phi_3}{\partial r}(r,s) = 0; & \dfrac{\partial \phi_3}{\partial s}(r,s) = 1;
\end{array}$$

B.2.2 6-Node 2D Linear Triangular Element

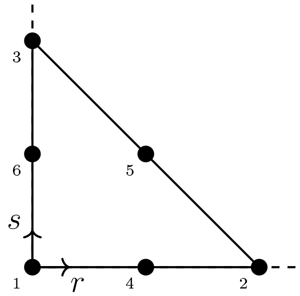


Figure B-4: Linear triangular element.

$$\begin{array}{lll}
\phi_1(r,s) = (1 - r - s)(1 - 2r - 2s); & \dfrac{\partial \phi_1}{\partial r}(r,s) = 4r + 4s - 3; & \dfrac{\partial \phi_1}{\partial s}(r,s) = 4r + 4s - 3; \\
\phi_2(r,s) = r(2r - 1); & \dfrac{\partial \phi_2}{\partial r}(r,s) = 4r - 1; & \dfrac{\partial \phi_2}{\partial s}(r,s) = 0; \\
\phi_3(r,s) = s(2s - 1); & \dfrac{\partial \phi_3}{\partial r}(r,s) = 0; & \dfrac{\partial \phi_3}{\partial s}(r,s) = 4s - 1; \\
\phi_4(r,s) = 4r(1 - r - s); & \dfrac{\partial \phi_4}{\partial r}(r,s) = 4 - 8r - 4s; & \dfrac{\partial \phi_4}{\partial s}(r,s) = -4r; \\
\phi_5(r,s) = 4rs; & \dfrac{\partial \phi_5}{\partial r}(r,s) = 4s; & \dfrac{\partial \phi_5}{\partial s}(r,s) = 4r; \\
\phi_6(r,s) = 4s(1 - r - s); & \dfrac{\partial \phi_6}{\partial r}(r,s) = -4s; & \dfrac{\partial \phi_6}{\partial s}(r,s) = 4 - 4r - 8s;
\end{array}$$

B.2.3 4-Node 2D Linear Rectangular Element

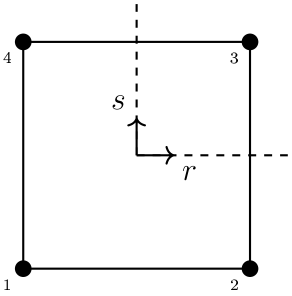


Figure B-5: Linear rectangular element.

The shape functions for the linear 4-node rectangular reference element and its partial derivatives with respect to the local coordinates r and s are:

$$\begin{aligned}
&\phi_1(r,s) = (1-r)(1-s)/4; && \frac{\partial\phi_1}{\partial r}(r,s) = -(1-s)/4; && \frac{\partial\phi_1}{\partial s}(r,s) = -(1-r)/4;\\
&\phi_2(r,s) = (1+r)(1-s)/4; && \frac{\partial\phi_2}{\partial r}(r,s) = (1-s)/4; && \frac{\partial\phi_2}{\partial s}(r,s) = -(1+r)/4;\\
&\phi_3(r,s) = (1+r)(1+s)/4; && \frac{\partial\phi_3}{\partial r}(r,s) = (1+s)/4; && \frac{\partial\phi_3}{\partial s}(r,s) = (1+r)/4;\\
&\phi_4(r,s) = (1-r)(1+s)/4. && \frac{\partial\phi_4}{\partial r}(r,s) = -(1+s)/4; && \frac{\partial\phi_4}{\partial s}(r,s) = (1-r)/4.
\end{aligned}$$

B.2.4 8-Node 2D Quadratic Rectangular Element

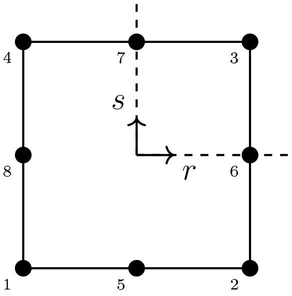


Figure B-6: Quadratic rectangular element.

The shape functions of the 8-node quadratic rectangular element are:

$$\phi_1 = (1-r)(1-s)(-r-s-1)/4; \qquad \phi_2 = (1+r)(1-s)(r-s-1)/4;$$

$$\phi_3 = (1+r)(1+s)(\quad r+s-1)/4; \qquad \phi_4 = (1-r)(1+s)(-r+s-1)/4;$$

$$\phi_5 = (1-r^2)(1-s)/2; \qquad \phi_6 = (1-s^2)(1+r)/2;$$

$$\phi_7 = (1-r^2)(1+s)/2; \qquad \phi_8 = (1-s^2)(1-r)/2.$$

The derivatives of the shape functions with respect to the local coordinate r are :

$$\frac{\partial \phi_1}{\partial r}(r,s) = (1-s)(2r+s)/4; \qquad \frac{\partial \phi_2}{\partial r}(r,s) = (1-s)(2r-s)/4;$$

$$\frac{\partial \phi_3}{\partial r}(r,s) = (1+s)(2r+s)/4; \qquad \frac{\partial \phi_4}{\partial r}(r,s) = (1+s)(2r-s)/4;$$

$$\frac{\partial \phi_5}{\partial r}(r,s) = -r(1-s); \qquad \frac{\partial \phi_6}{\partial r}(r,s) = (1-s^2)/2;$$

$$\frac{\partial \phi_7}{\partial r}(r,s) = -r(1+s); \qquad \frac{\partial \phi_8}{\partial r}(r,s) = -(1-s^2)/2$$

The derivatives of the shape functions with respect to the local coordinate s are :

$$\frac{\partial \phi_1}{\partial s}(r,s) = (1-r)(2s+r)/4; \qquad \frac{\partial \phi_1}{\partial s}(r,s) = (1+r)(2s-r)/4;$$

$$\frac{\partial \phi_1}{\partial s}(r,s) = (1+r)(2s+r)/4; \qquad \frac{\partial \phi_1}{\partial s}(r,s) = (1-r)(2s-r)/4;$$

$$\frac{\partial \phi_1}{\partial s}(r,s) = (r^2-1)/2; \qquad \frac{\partial \phi_1}{\partial s}(r,s) = -s(1+r);$$

$$\frac{\partial \phi_1}{\partial s}(r,s) = (1-r^2)/2; \qquad \frac{\partial \phi_1}{\partial s}(r,s) = -s(1-r);$$

B.3 Shape Functions for 3D Elements

B.3.1 4-Node 3D Tetrahedral Element (C3D4)

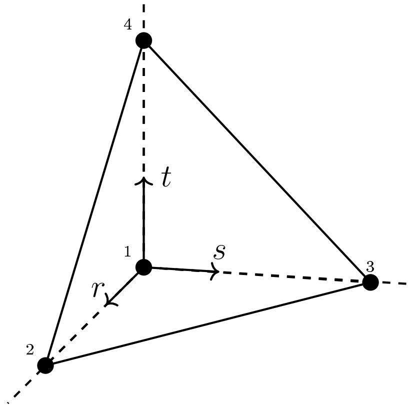


Figure B-7: Linear tetrahedral element.

The shape functions for the linear 4-node tetrahedral reference element and its partial derivatives with respect to the local coordinates r, s and t are:

$$
\begin{aligned}
&\phi_1(r,s,t) = 1 - r - s - t; && \frac{\partial \phi_1}{\partial r} = -1; && \frac{\partial \phi_1}{\partial s} = -1; && \frac{\partial \phi_1}{\partial t} = -1;\\
&\phi_2(r,s,t) = r; && \frac{\partial \phi_2}{\partial r} = 1; && \frac{\partial \phi_2}{\partial s} = 0; && \frac{\partial \phi_2}{\partial t} = 0;\\
&\phi_3(r,s,t) = s; && \frac{\partial \phi_3}{\partial r} = 0; && \frac{\partial \phi_3}{\partial s} = 1; && \frac{\partial \phi_3}{\partial t} = 0;\\
&\phi_4(r,s,t) = t; && \frac{\partial \phi_4}{\partial r} = 0; && \frac{\partial \phi_4}{\partial s} = 0; && \frac{\partial \phi_4}{\partial t} = 1.
\end{aligned}
$$

B.3.2 10-Node 3D Tetrahedral Element (C3D10)

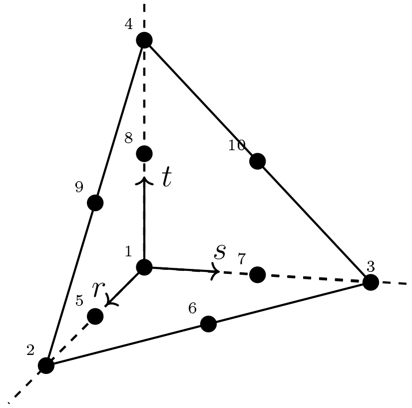


Figure B-8: Quadratic tetrahedral element.

The shape functions for the quadratic 4-node tetrahedral reference element and its partial derivatives with respect to the local coordinates r, s and t are:

$$\begin{aligned}
\phi_1(r,s,t) &= (1-r-s-t)(1-2(r+s+t)); & \frac{\partial \phi_1}{\partial r}(r,s,t) &= 4(r+s+t)-3;\\
\phi_2(r,s,t) &= (2r-1)r; & \frac{\partial \phi_2}{\partial r}(r,s,t) &= 4r-1;\\
\phi_3(r,s,t) &= (2s-1)s; & \frac{\partial \phi_3}{\partial r}(r,s,t) &= 0;\\
\phi_4(r,s,t) &= (2t-1)t; & \frac{\partial \phi_4}{\partial r}(r,s,t) &= 0;\\
\phi_5(r,s,t) &= 4r(1-r-s-t); & \frac{\partial \phi_5}{\partial r}(r,s,t) &= 4(1-2r-s-t);\\
\phi_6(r,s,t) &= 4rs; & \frac{\partial \phi_6}{\partial r}(r,s,t) &= 4s;\\
\phi_7(r,s,t) &= 4s(1-r-s-t); & \frac{\partial \phi_7}{\partial r}(r,s,t) &= -4s;\\
\phi_8(r,s,t) &= 4t(1-r-s-t); & \frac{\partial \phi_8}{\partial r}(r,s,t) &= -4t;\\
\phi_9(r,s,t) &= 4rt; & \frac{\partial \phi_9}{\partial r}(r,s,t) &= 4t;\\
\phi_{10}(r,s,t) &= 4st; & \frac{\partial \phi_{10}}{\partial r}(r,s,t) &= 0;
\end{aligned}$$

$$\frac{\partial \phi_1}{\partial s}(r,s,t) = 4(r+s+t)-3; \qquad \frac{\partial \phi_1}{\partial t}(r,s,t) = 4(r+s+t)-3;$$
$$\frac{\partial \phi_2}{\partial s}(r,s,t) = 0; \qquad \frac{\partial \phi_2}{\partial t}(r,s,t) = 0;$$
$$\frac{\partial \phi_3}{\partial s}(r,s,t) = 4s-1; \qquad \frac{\partial \phi_3}{\partial t}(r,s,t) = 0;$$
$$\frac{\partial \phi_4}{\partial s}(r,s,t) = 0; \qquad \frac{\partial \phi_4}{\partial t}(r,s,t) = 4t-1;$$
$$\frac{\partial \phi_5}{\partial s}(r,s,t) = -4r; \qquad \frac{\partial \phi_5}{\partial t}(r,s,t) = -4r;$$
$$\frac{\partial \phi_6}{\partial s}(r,s,t) = 4r; \qquad \frac{\partial \phi_6}{\partial t}(r,s,t) = 0;$$
$$\frac{\partial \phi_7}{\partial s}(r,s,t) = 4(1-r-2s-t); \qquad \frac{\partial \phi_7}{\partial t}(r,s,t) = -4s;$$
$$\frac{\partial \phi_8}{\partial s}(r,s,t) = -4t; \qquad \frac{\partial \phi_8}{\partial t}(r,s,t) = 4(1-r-s-2t);$$
$$\frac{\partial \phi_9}{\partial s}(r,s,t) = 0; \qquad \frac{\partial \phi_9}{\partial t}(r,s,t) = 4r;$$
$$\frac{\partial \phi_{10}}{\partial s}(r,s,t) = 4t; \qquad \frac{\partial \phi_{10}}{\partial t}(r,s,t) = 4s;$$

B.3.3 8-Node 3D Isoparametric Element (C3D8)

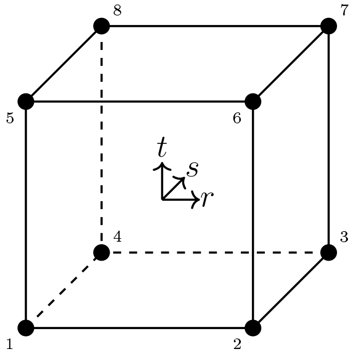


Figure B-9: Linear brick element.

The shape functions for the linear 8-node brick reference element are given as follow:

$$\begin{aligned}
\phi_1(r,s,t) &= (1-r)(1-s)(1-t)/8; & \phi_2(r,s,t) &= (1+r)(1-s)(1-t)/8;\\
\phi_3(r,s,t) &= (1+r)(1+s)(1-t)/8; & \phi_4(r,s,t) &= (1-r)(1+s)(1-t)/8;\\
\phi_5(r,s,t) &= (1-r)(1-s)(1+t)/8; & \phi_6(r,s,t) &= (1+r)(1-s)(1+t)/8;\\
\phi_7(r,s,t) &= (1+r)(1+s)(1+t)/8; & \phi_8(r,s,t) &= (1-r)(1+s)(1+t)/8;
\end{aligned}$$

The partial derivatives of the above 8-node element with respect to r, s and t are given as follow:

$$\begin{aligned}
\frac{\partial\phi_1}{\partial r}(r,s,t) &= -(1-s)(1-t)/8; & \frac{\partial\phi_2}{\partial r}(r,s,t) &= \quad(1-s)(1-t)/8;\\
\frac{\partial\phi_3}{\partial r}(r,s,t) &= \quad(1+s)(1-t)/8; & \frac{\partial\phi_4}{\partial r}(r,s,t) &= -(1+s)(1-t)/8;\\
\frac{\partial\phi_5}{\partial r}(r,s,t) &= -(1-s)(1+t)/8; & \frac{\partial\phi_6}{\partial r}(r,s,t) &= \quad(1-s)(1+t)/8;\\
\frac{\partial\phi_7}{\partial r}(r,s,t) &= \quad(1+s)(1+t)/8; & \frac{\partial\phi_8}{\partial r}(r,s,t) &= -(1+s)(1+t)/8;
\end{aligned}$$

$$\begin{aligned}
\frac{\partial\phi_1}{\partial s}(r,s,t) &= -(1-r)(1-t)/8; & \frac{\partial\phi_2}{\partial s}(r,s,t) &= -(1+r)(1-t)/8;\\
\frac{\partial\phi_3}{\partial s}(r,s,t) &= \quad(1+r)(1-t)/8; & \frac{\partial\phi_4}{\partial s}(r,s,t) &= \quad(1-r)(1-t)/8;\\
\frac{\partial\phi_5}{\partial s}(r,s,t) &= -(1-r)(1+t)/8; & \frac{\partial\phi_6}{\partial s}(r,s,t) &= -(1+r)(1+t)/8;\\
\frac{\partial\phi_7}{\partial s}(r,s,t) &= \quad(1+r)(1+t)/8; & \frac{\partial\phi_8}{\partial s}(r,s,t) &= \quad(1-r)(1+t)/8;
\end{aligned}$$

$$\begin{aligned}
\frac{\partial\phi_1}{\partial t}(r,s,t) &= -(1-r)(1-s)/8; & \frac{\partial\phi_2}{\partial t}(r,s,t) &= -(1+r)(1-s)/8;\\
\frac{\partial\phi_3}{\partial t}(r,s,t) &= -(1+r)(1+s)/8; & \frac{\partial\phi_4}{\partial t}(r,s,t) &= -(1-r)(1+s)/8;\\
\frac{\partial\phi_5}{\partial t}(r,s,t) &= \quad(1-r)(1-s)/8; & \frac{\partial\phi_6}{\partial t}(r,s,t) &= \quad(1+r)(1-s)/8;\\
\frac{\partial\phi_7}{\partial t}(r,s,t) &= \quad(1+r)(1+s)/8; & \frac{\partial\phi_8}{\partial t}(r,s,t) &= \quad(1-r)(1+s)/8;
\end{aligned}$$

B.3.4 20-Node 3D Isoparametric Element (C3D20)

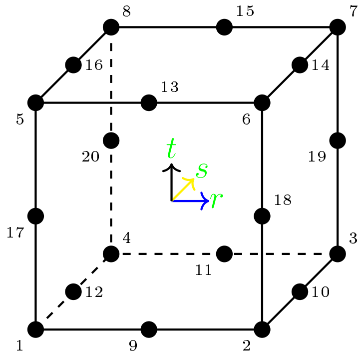


Figure B-10: Quadratic brick element.

The shape functions for the quadratic 20-node cubic element are given as follow:

$$\phi_1(r,s,t) = (1-r)(1-s)(1-t)(-r-s-t-2)/8;$$

$$\phi_2(r,s,t) = (1+r)(1-s)(1-t)(+r-s-t-2)/8;$$

$$\phi_3(r,s,t) = (1+r)(1+s)(1-t)(+r+s-t-2)/8;$$

$$\phi_4(r,s,t) = (1-r)(1+s)(1-t)(-r+s-t-2)/8;$$

$$\phi_5(r,s,t) = (1-r)(1-s)(1+t)(-r-s+t-2)/8;$$

$$\phi_6(r,s,t) = (1+r)(1-s)(1+t)(+r-s+t-2)/8;$$

$$\phi_7(r,s,t) = (1+r)(1+s)(1+t)(+r+s+t-2)/8;$$

$$\phi_8(r,s,t) = (1-r)(1+s)(1+t)(-r+s+t-2)/8;$$

$$\phi_9(r,s,t) = (1+r)(1-r)(1-s)(1-t)/4; \qquad \phi_{10}(r,s,t) = (1+r)(1+s)(1-s)(1-t)/4;$$

$$\phi_{11}(r,s,t) = (1+r)(1-r)(1+s)(1-t)/4; \qquad \phi_{12}(r,s,t) = (1-r)(1+s)(1-s)(1-t)/4;$$

$$\phi_{13}(r,s,t) = (1+r)(1-r)(1-s)(1+t)/4; \qquad \phi_{14}(r,s,t) = (1+r)(1+s)(1-s)(1+t)/4;$$

$$\phi_{15}(r,s,t) = (1+r)(1-r)(1+s)(1+t)/4; \qquad \phi_{16}(r,s,t) = (1-r)(1+s)(1-s)(1+t)/4;$$

$$\phi_{17}(r,s,t) = (1-r)(1-s)(1+t)(1-t)/4; \qquad \phi_{18}(r,s,t) = (1+r)(1-s)(1+t)(1-t)/4;$$

$$\phi_{19}(r,s,t) = (1+r)(1+s)(1+t)(1-t)/4; \qquad \phi_{20}(r,s,t) = (1-r)(1+s)(1+t)(1-t)/4;$$

The partial derivatives of the above 20-node cubic element are given as follow:

$$\begin{aligned}
\frac{\partial \phi_1}{\partial r}(r,s,t) &= \quad (1-s)(1-t)(2r+s+t+1)/8;\\
\frac{\partial \phi_2}{\partial r}(r,s,t) &= \quad (1-s)(1-t)(2r-s-t-1)/8;\\
\frac{\partial \phi_3}{\partial r}(r,s,t) &= \quad (1+s)(1-t)(2r+s-t-1)/8;\\
\frac{\partial \phi_4}{\partial r}(r,s,t) &= \quad (1+s)(1-t)(2r-s+t+1)/8;\\
\frac{\partial \phi_5}{\partial r}(r,s,t) &= \quad (1-s)(1+t)(2r+s-t+1)/8;\\
\frac{\partial \phi_6}{\partial r}(r,s,t) &= \quad (1-s)(1+t)(2r-s+t-1)/8;\\
\frac{\partial \phi_7}{\partial r}(r,s,t) &= \quad (1+s)(1+t)(2r+s+t-1)/8;\\
\frac{\partial \phi_8}{\partial r}(r,s,t) &= \quad (1+s)(1+t)(2r-s-t+1)/8;
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \phi_9}{\partial r}(r,s,t) &= -r(1-s)(1-t)/2; & \frac{\partial \phi_{10}}{\partial r}(r,s,t) &= \quad (1+s)(1-s)(1-t)/4;\\
\frac{\partial \phi_{11}}{\partial r}(r,s,t) &= -r(1+s)(1-t)/2; & \frac{\partial \phi_{12}}{\partial r}(r,s,t) &= -(1+s)(1-s)(1-t)/4;\\
\frac{\partial \phi_{13}}{\partial r}(r,s,t) &= -r(1-s)(1+t)/2; & \frac{\partial \phi_{14}}{\partial r}(r,s,t) &= \ (1+s)(1-s)(1+t)/4;\\
\frac{\partial \phi_{15}}{\partial r}(r,s,t) &= -r(1+s)(1+t)/2; & \frac{\partial \phi_{16}}{\partial r}(r,s,t) &= -(1+s)(1-s)(1+t)/4;\\
\frac{\partial \phi_{17}}{\partial r}(r,s,t) &= \ -(1-s)(1+t)(1-t)/4; & \frac{\partial \phi_{18}}{\partial r}(r,s,t) &= \quad (1-s)(1+t)(1-t)/4;\\
\frac{\partial \phi_{19}}{\partial r}(r,s,t) &= \quad (1+s)(1+t)(1-t)/4; & \frac{\partial \phi_{20}}{\partial r}(r,s,t) &= \quad (1+s)(1+t)(1-t)/4;
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \phi_1}{\partial s}(r,s,t) &= (1-r)(1-t)(2s+r+t+1)/8;\\
\frac{\partial \phi_2}{\partial s}(r,s,t) &= (1+r)(1-t)(2s-r+t+1)/8;\\
\frac{\partial \phi_3}{\partial s}(r,s,t) &= (1+r)(1-t)(2s+r-t-1)/8;\\
\frac{\partial \phi_4}{\partial s}(r,s,t) &= (1-r)(1-t)(2s-r-t-1)/8;\\
\frac{\partial \phi_5}{\partial s}(r,s,t) &= (1-r)(1+t)(2s+r-t+1)/8;\\
\frac{\partial \phi_6}{\partial s}(r,s,t) &= (1+r)(1+t)(2s-r-t+1)/8;\\
\frac{\partial \phi_7}{\partial s}(r,s,t) &= (1+r)(1+t)(2s+r+t-1)/8;\\
\frac{\partial \phi_8}{\partial s}(r,s,t) &= (1-r)(1+t)(2s-r+t-1)/8;
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \phi_9}{\partial s}(r,s,t) &= -(1+r)(1-r)(1-t)/4; & \frac{\partial \phi_{10}}{\partial s}(r,s,t) &= -(1+r)s(1-t)/2;\\
\frac{\partial \phi_{11}}{\partial s}(r,s,t) &= \quad (1+r)(1-r)(1-t)/4; & \frac{\partial \phi_{12}}{\partial s}(r,s,t) &= -(1-r)s(1-t)/2;\\
\frac{\partial \phi_{13}}{\partial s}(r,s,t) &= -(1+r)(1-r)(1+t)/4; & \frac{\partial \phi_{14}}{\partial s}(r,s,t) &= -(1+r)s(1+t)/2;\\
\frac{\partial \phi_{15}}{\partial s}(r,s,t) &= \quad (1+r)(1-r)(1+t)/4; & \frac{\partial \phi_{16}}{\partial s}(r,s,t) &= -(1-r)s(1+t)/2;\\
\frac{\partial \phi_{17}}{\partial s}(r,s,t) &= -(1-r)(1+t)(1-t)/4; & \frac{\partial \phi_{18}}{\partial s}(r,s,t) &= -(1+r)(1+t)(1-t)/4;\\
\frac{\partial \phi_{19}}{\partial s}(r,s,t) &= \quad (1+r)(1+t)(1-t)/4; & \frac{\partial \phi_{20}}{\partial s}(r,s,t) &= \quad (1-r)(1+t)(1-t)/4;
\end{aligned}$$

$$\begin{aligned}
\frac{\partial\phi_1}{\partial t}(r,s,t) &= (1-s)(1-r)(2t+r+s+1)/8;\\
\frac{\partial\phi_2}{\partial t}(r,s,t) &= (1+r)(1-s)(2t-r+s+1)/8;\\
\frac{\partial\phi_3}{\partial t}(r,s,t) &= (1+r)(1+s)(2t-r-s+1)/8;\\
\frac{\partial\phi_4}{\partial t}(r,s,t) &= (1-r)(1+s)(2t+r-s+1)/8;\\
\frac{\partial\phi_5}{\partial t}(r,s,t) &= (1-r)(1-s)(2t-r-s-1)/8;\\
\frac{\partial\phi_6}{\partial t}(r,s,t) &= (1+r)(1-s)(2t+r-s-1)/8;\\
\frac{\partial\phi_7}{\partial t}(r,s,t) &= (1+r)(1+s)(2t+r+s-1)/8;\\
\frac{\partial\phi_8}{\partial t}(r,s,t) &= (1-r)(1+s)(2t-r+s-1)/8;
\end{aligned}$$

$$\begin{aligned}
\frac{\partial\phi_9}{\partial t}(r,s,t) &= -(1+r)(1-r)(1-s)/4; & \frac{\partial\phi_{10}}{\partial t}(r,s,t) &= -(1+r)(1+s)(1-s)/4;\\
\frac{\partial\phi_{11}}{\partial t}(r,s,t) &= -(1+r)(1-r)(1+s)/4; & \frac{\partial\phi_{12}}{\partial t}(r,s,t) &= -(1-r)(1+s)(1-s)/4;\\
\frac{\partial\phi_{13}}{\partial t}(r,s,t) &= (1+r)(1-r)(1-s)/4; & \frac{\partial\phi_{14}}{\partial t}(r,s,t) &= (1+r)(1+s)(1-s)/4;\\
\frac{\partial\phi_{15}}{\partial t}(r,s,t) &= (1+r)(1-r)(1+s)/4; & \frac{\partial\phi_{16}}{\partial t}(r,s,t) &= (1-r)(1+s)(1-s)/4;\\
\frac{\partial\phi_{17}}{\partial t}(r,s,t) &= -(1-r)(1-s)t/2; & \frac{\partial\phi_{18}}{\partial t}(r,s,t) &= -(1+r)(1-s)t/2;\\
\frac{\partial\phi_{19}}{\partial t}(r,s,t) &= -(1+r)(1+s)t/2; & \frac{\partial\phi_{20}}{\partial t}(r,s,t) &= -(1-r)(1+s)t/2;
\end{aligned}$$

Appendix C

Sensitivity Computation

C.0.1 Linear Elasticity

To compute the Sensitivity $\frac{dJ(X,U(X))}{dX}$, where X is a vector of nodes, we use in this section the adjoint method. This technique allows us to evaluate efficiently the partial derivatives with respect to many control parameters.

$$\frac{dJ(X,U(X))}{dX} = \frac{\partial J(X,U(X))}{\partial U}\frac{\partial U}{\partial X} + \frac{\partial J(X,U(X))}{\partial X} \tag{C.1}$$

The residual $R(X,U(X))$ is defined as:

$$R(X,U(X)) = B(X,U(X)) - F(X)$$

After deriving R with respect to X we obtain:

$$\frac{dR}{dX} = \frac{\partial R}{\partial U}\frac{\partial U}{\partial X} + \frac{\partial R}{\partial X} = 0,$$

which implies that:

$$\frac{\partial R}{\partial U}\frac{\partial U}{\partial X} = -\frac{\partial R}{\partial X},$$

and thus

$$\frac{\partial U}{\partial X} = -\left(\frac{\partial R}{\partial U}\right)^{-1} \frac{\partial R}{\partial X} \tag{C.2}$$

By substituting C.2 in C.1 we have:

$$\begin{aligned} \frac{dJ}{dX} &= \frac{\partial J}{\partial U}\frac{\partial U}{\partial X} + \frac{\partial J}{\partial X} \\ &= -\frac{\partial J}{\partial U}\left(\frac{\partial R}{\partial U}\right)^{-1} \frac{\partial R}{\partial X} + \frac{\partial J}{\partial X} \end{aligned}$$

Now we solve the following adjoint problem:

$$\begin{cases} \text{solve } (\frac{\partial R}{\partial U})^T \lambda_1 = (\frac{\partial J}{\partial U})^T \\ \text{and compute } \frac{dJ}{dX} = -\lambda_1^T \frac{\partial R}{\partial X} + \frac{\partial J}{\partial X} \end{cases} \tag{C.3}$$

where $\lambda_1 = (\frac{\partial J}{\partial U}(\frac{\partial R}{\partial U})^{-1})^T$.

C.0.2 Linear Thermoelasticity

We return to the algebraic thermo-mechanical system described in the section 5.3:

$$\begin{cases} B_M U + B_{MT} T = F_M \\ B_T U = F_T \end{cases} .$$

We set the residuals R_M and R_T as:

$$\begin{cases} R_M = B_M U + B_{MT} T - F_M \\ R_T = B_T U - F_T \end{cases}$$

From the second equation we can write:

$$\frac{dR_T}{dX} = \frac{\partial R_T}{\partial T}\frac{\partial T}{\partial X} + \frac{\partial R_T}{\partial X} = 0,$$

it follows

$$\frac{\partial T}{\partial X} = -\Big(\frac{\partial R_T}{\partial T}\Big)^{-1}\frac{\partial R_T}{\partial X}$$

From the first equation we can write:

$$\frac{dR_M}{dX} = \frac{\partial R_M}{\partial u}\frac{\partial u}{\partial X} + \frac{\partial R_M}{\partial T}\frac{\partial T}{\partial X} + \frac{\partial R_M}{\partial X} = 0$$

it follows:

$$\begin{aligned}\frac{\partial u}{\partial X} &= -\Big(\frac{\partial R_M}{\partial u}\Big)^{-1}\Big[\frac{\partial R_M}{\partial T}\frac{\partial T}{\partial X} + \frac{\partial R_M}{\partial X}\Big]\\ &= -\Big(\frac{\partial R_M}{\partial u}\Big)^{-1}\Big[-\frac{\partial R_M}{\partial T}\Big(\frac{\partial R_T}{\partial T}\Big)^{-1}\frac{\partial R_T}{\partial X} + \frac{\partial R_M}{\partial X}\Big]\end{aligned}$$

The global thermal shape sensitivity is:

$$\begin{aligned}\frac{dJ}{dX} &= \frac{\partial J}{\partial u}\frac{\partial u}{\partial X} + \frac{\partial J}{\partial T}\frac{\partial T}{\partial X} + \frac{\partial J}{\partial X}\\ &= -\frac{\partial J}{\partial u}\Big(\frac{\partial R_M}{\partial u}\Big)^{-1}\Big[-\frac{\partial R_M}{\partial T}\Big(\frac{\partial R_T}{\partial T}\Big)^{-1}\frac{\partial R_T}{\partial X} + \frac{\partial R_M}{\partial X}\Big]\\ &\quad -\frac{\partial J}{\partial T}\Big(\frac{\partial R_T}{\partial T}\Big)^{-1}\frac{\partial R_T}{\partial X} + \frac{\partial J}{\partial X}.\end{aligned}$$

It follows

$$\begin{aligned}\frac{dJ}{dX} &= \overbrace{\Big[\frac{\partial J}{\partial u}\Big(\frac{\partial R_M}{\partial u}\Big)^{-1}\frac{\partial R_M}{\partial T} - \frac{\partial J}{\partial T}\Big]\Big(\frac{\partial R_T}{\partial T}\Big)^{-1}}^{\lambda_2^T}\frac{\partial R_T}{\partial X}\\ &\quad -\underbrace{\frac{\partial J}{\partial u}\Big(\frac{\partial R_M}{\partial u}\Big)^{-1}}_{\lambda_1^T}\frac{\partial R_M}{\partial X} + \frac{\partial J}{\partial X}\end{aligned}$$

We set

$$\begin{cases} \lambda_2^T = \left[\frac{\partial J}{\partial u}\left(\frac{\partial R_M}{\partial u}\right)^{-1}\frac{\partial R_M}{\partial T} - \frac{\partial J}{\partial T}\right]\left(\frac{\partial R_T}{\partial T}\right)^{-1} & \text{and} \\ \lambda_1^T = \frac{\partial J}{\partial u}\left(\frac{\partial R_M}{\partial u}\right)^{-1} & . \end{cases}$$

Now we have to solve the following problem:

$$\begin{cases} \text{solve} \quad \overbrace{\left(\frac{\partial R_M}{\partial u}\right)^T \lambda_1 = \left(\frac{\partial J}{\partial u}\right)^T}^{purely\ mechanical\ adjoint\ equation} \\ \\ \text{and} \ \overbrace{\left(\frac{\partial R_T}{\partial T}\right)^T \lambda_2 = \left(\lambda_1^T \frac{\partial R_M}{\partial T} - \frac{\partial J}{\partial T}\right)^T}^{thermal-mechanical\ adjoint\ equation} \\ \\ \text{and compute} \ \underbrace{\frac{dJ}{dX} = \lambda_2^T \frac{\partial R_T}{\partial X} - \lambda_1^T \frac{\partial R_M}{\partial X} + \frac{\partial J}{\partial X}}_{global\ sensitivity\ equation} \end{cases}$$

Bibliography

[1] S. Agmon, A. Douglis and L. Nirenberg, *Estimates Near the Boundary for Solutions of Elliptic Partial Differential Equations Satisfying General Boundary Conditions II*, Communications On Pure And Applied Mathematics, Vol. XVII, pp. 35-92, 1964.

[2] G. Allaire, *Numerical Analysis and Optimization*, Oxford University Press, Oxford (2007)

[3] G. Allaire, E. Bonneter, G. Francfort and F. Jouve, *Shape Optimization by the Homogenization Method*, Numer. Math., **76**, 1997, 27-68.

[4] W. Bangerth and R. Rannacher, *Adaptive Finite Element Methods for Differential Equations*, Birkhäuser 2003.

[5] R. E. Bank, J.-C. Xu and B. Zheng, *Superconvergent Derivative Recovery for Lagrange Triangular Elements of Degree p on Unstructured Grids*, SIAM J. Numer. Anal. Vol. 45, 2007, 2032-2046.

[6] M. Bäker, H. Harders and J. Rösler, *Mechanical Behaviour of Engineering Materials: Metals, Ceramics, Polymers, and Composites*, German edition published by Teubner Verlag (Wiesbaden, 2006), Springer, Berlin Heidelberg 2007.

[7] S. B. Batdorf and J. G. Crosse, *A Statistical Theory for the Fracture of Brittle Structures Subject to Nonuniform Polyaxial Stress*, J. Appl. Mech. 41, 459–465 (1974)

[8] M. Bolten, H. Gottschalk, C. Hahn and M. Saadi, *Numerical Shape Optimization To Decrease Failure Probability Of Ceramic Structures*, Preprint 2017, arXiv:1705.05776.

[9] M. Bolten, H. Gottschalk and S. Schmitz, *Minimal Failure Probability for Ceramic Design via Shape Control*, J. Optim. Theory Appl., **166** (2015), 983–1001.

[10] L.P. Borrego, L.M. Abreu, J.M. Costa and J.M. Ferreira, *Analysis of Low Cycle Fatigue in AlMgSi Aluminium Alloys*, Engineering Failure Analysis **11** (2004) 715-725.

[11] A. Borzi and V. Schulz, *Computational Optimization of Systems governed by Partial Differential Equations*, SIAM series on computational engineering, SIAM 2012.

[12] D. Braess, *Finite Elemente - Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie*, fourth edition, Springer, Berlin, 2007.

[13] P. Ciarlet, *Mathematical Elasticity - Volume I: Three-Dimensional Elasticity*, Studies in Mathematics and its Applications, Vol. 20, North-Holland, Amsterdam, 1988.

[14] P. Ciarlet, *Basic Error Estimates for Elliptic Problems, Vol. II: Finite Element Methods*, Handbook of Numerical Analysis, Vol. 20,P.G. Ciarlet and J.-L. Lions, North-Holland, Amsterdam, 1991.

[15] P. Ciarlet, *The Finite Element Method for Elliptic Problems*, Studies in Mathematics and its Applications, Elsevier Science, 1978.

[16] CARL C. OSGOOD, *Fatigue Design*, Library of Congress Cataloging in Publication Data, New Jersey, USA 1982.

[17] A. Ern and J.-L. Guermond, *Theory and Practice of Finite Elements*, Springer, New York, 2004.

[18] L. A. Escobar and W. Q. Meeker, *Statistical Methods for Reliability Data*, Wiley-Interscience Publication, New York, 1998.

[19] D. Munz and T. Fett, *Ceramics*, (engl. edition) Springer Verlag Berlin, 2012.

[20] C. Frey, D. Nürnberger and H.-P. Kersken, *The Discrete Adjoint of a Turbomashinery RANS Solver*, ASME Turbo expo 2009, GT2009-59062

[21] H. Gottschalk and S. Schmitz, *Optimal Reliability in Design for Fatigue Life I: Existence of optimal shapes*, SIAM J. Control Optim. **52** Vol. 5, (2014), 2727-2752.

[22] H. Gottschalk, S. Schmitz, T. Seibel, R. Krause, G. Rollmann and T. Beck, *Probabilistic Schmid Factors and Scatter of LCF Life* , Preprint 2014, to appear in Materials Science and Engineering.

[23] G. Gottstein, *Physical Foundations of Material Science*, Springer-Verlag Berlin Heidelberg 2004.

[24] J. Haslinger and R. A. E. Mäkinen, *Introduction to Shape Optimization - Theory, Approximation and Computation*, SIAM - Advances in Design and Control, 2003.

[25] R. B. Hetnarski and M. R. Eslami, *Thermal Stresses - Advanced Theory and Applications.*, Springer, 2010.

[26] O. Hertel, and M. Vormwald, *Statistical and Geometrical Size Effects in Notched Members Based on Weakest-link and Short-crack Modelling*, Engineering Fracture Mechanics, Volume 95, November 2012, Pages 72-83.

[27] M. Hoffmann and T. Seeger, *A Generalized Method for Estimating Elastic-Plastic Notch Stresses and Strains, Part 1: Theory*, Journal of Engineering Materials and Technology, 107, pp. 250-254, 1985.

[28] J. C. Wehrstedt, *Shape Optimzation with Variatioal Inequalities as Constraint and an Application in Pine Surgery* (in German), PhD Thesis in Mathematics, Technical University, Munchen 2007.

[29] J. MARTINS, J. ALONSO and J. REUTHER, *A Coupled-Adjoint Sensitivity Analysis Method for High-Fidelity Aero-Structural Design Optimization and Engineering*, 6, 33–62, 2005, Springer Science, 2005.

[30] O. Kallenberg, *Random Measures*, Akademie-Verlag, Berlin 1983.

[31] M. Knop, R. Jones, L. Molent and L. Wang, *On Glinka and Neuber Methods for Calculating Notch Tip Strains under Cyclic Load Spectra*, International Journal of Fatigue, Vol. 22, (2000) 743–755.

[32] S. Kämmerer, J. F. Mayer, M. Paffrath, U. Wever and A. R. Jung, *Three Dimensional Optimization of Turbomashinery Blading using Sensitivity Analysis*, ASME Turbo expo 2003, GT2003-38037

[33] A. Keskin, M. Swoboda, P.M. Flassing, A. K. Dutta and D. Bestle, *Accelerated Industrial Blade Design Based on Multi-objective Optimization using Surrogate Model Methodology*, ASME Turbo expo 2008, GT2008-50506

[34] W. B. Liu, P. Neittaanmäki and D. Tiba, *Existence for Shape Optimization Problems in Arbitrary Dimension*, SIAM J. Contr. Optimization **41** (2003) 1440-1454.

[35] G. R. Leverant, D. L. Littlefield, R. C. McClung, H. R. Millwater, and J. Y. Wu, *A Probabilistic Approach to Aircraft Turbine Rotor Material Design*, Paper 97-GT-22, ASME Turbo Expo '97, Orlando, Florida, June 1997.

[36] L. Maede, S. Schmitz, H. Gottschalk and T. Beck, *Combined Notch and Size Effect Modeling in a Local Probabilistic Approach for LCF*, Computational Materials Science Volume 142, 1 February 2018, Pages 377-388

[37] J. Luo, J. Xiong, F. Liu and I. McBean, *Secondary Flow Reduction by Blade Redesign and Endwall Contouring using an Adjoint Optimization Method*, ASME Turbo expo 2010, GT2010-22061

[38] Max D. Gunzburger, *Perspectives in Flow Control and Optimization*, Society for Industrial and Applied Mathematics, Januar 1987

[39] H. Neuber, *Theory of Stress Concentration for Shear-Strained Prismatical Bodies with Arbitrary Nonlinear Stress-Strain Law*, J. Appl. Mech. 26, 544, 1961.

[40] D. Radaj and M. Vormwald, *Ermüdungsfestigkeit*, third edition, Springer, Berlin Heidelberg, 2007.

[41] W. Ramberg and W. R. Osgood, *Description of Stress-Strain Curves by Three Parameters*, Technical Notes - National Advisory Committee For Aeronautics, No. 902, Washington DC., 1943.

[42] H. Riesch-Oppermann, A. Brückner-Foit and C. Ziegler, *STAU - A General Purpose Tool for Probabilistic Reliability Assessment of Ceramic Components under Multi Axial Loading*, in: Proceedings of the 13th International Conference on ECF 13, San Sebastian (2000)

[43] S. Schmitz, *A Probabilistic Local Model for Low Cycle Fatigue – New Aspects of Structurel Mechanics*, Dissertation Lugano and Wuppertal 2014, appeared in Hartung-Gorre Verlag, 2014.

[44] S. Schmitz, G. Rollmann, H. Gottschalk and R. Krause, *Risk Estimation for LCF Crack Initiation*, Proc. ASME Turbo Expo 2013, GT2013-94899, arXiv:1302.2909v1.

[45] S. Schmitz, G. Rollmann, H. Gottschalk and R. Krause, *Probabilistic Analysis of the LCF Crack Initiation Life for a Turbine Blade under Thermo-mechanical Loading*, Proc. Int. Conf. LCF 7 (September 13).

[46] S. Schmitz, T. Seibel, T. Beck, G. Rollmann, R. Krause and H. Gottschalk, *A Probabilistic Model for LCF*, Computational Materials Science **79** (2013), 584-590.

[47] J. Sokolowski and J.-P. Zolesio, *Introduction to Shape Optimization - Shape Sensivity Analysis*, first edition, Springer, Berlin Heidelberg, 1992.

[48] D. Sornette, T. Magnin, and Y. Brechet, *The Physical Origin of the Coffin-Manson Law in Low-Cycle Fatigue*, Europhys. Lett., 20 (1992), pp. 433–438.

[49] F. Tröltzsch, *Optimal Control of Partial Differential Equations (in German)*, Vieweg + Teubner, 2010.

[50] S. Watanabe, *On Discontinuous Additive Functionals and Lévy Measures of a Markov Process*, Japan. J. Math. 34 (1964).

[51] E. W. Weibull, *A Statistical Theory of the Strength of Materials*, Ingeniors Vetenskaps Akad. Handl. 151, 1–45 (1939)

[52] K. Wittig, *Construction of a Gas Turbine for Model Air Planes (in German)*, Munich 1993, www.calculix.de.

[53] D. X. Wang and Y. S. Li, *3D Direct and Inverse Design using NS Equations and the Adjoint Method for Turbine Blades*, ASME Turbo Expo 2010, GT2010-22049

[54] A. Nowacki, *Thermoelasticity*, Pergamon Press, International Series of Monographs in Aeronautics and Astronautics (1962).

[55] J. Prevost and D. Tao, *Finite Element Analysis of Dynamic Coupled Thermoelasticity Problems With Relaxation Times*, Journal of Applied Mechanics-transactions of The Asme - J APPL MECH, Vol. 50, 1983/12/01

[56] E. Ellobody, *Finite Element Analysis and Design of Steel and Steel-Concrete Composite Bridges*, Butterworth-Heinemann, 2014

[57] D. Gross and T. Seelig. *Bruchmechanik*, Springer-Verlag Berlin Heidelberg, 2007.

[58] T. Beck, H. Gottschalk, R. Krause, G. Rollmann, S. Schmitz and T. Seibel, *Probabilistische Lebensdauerberechnung für Design bei extremen Temperaturen : Teilverbundprojekt "Expansion" ; Vorhabengruppe 4.1 "Gas- und Dampfturbinenschaufeln"*, Technische Informationsbibliothek u. Universitätsbibliothek, 2013, Reportnr. : 0327718A

[59] C. Kontermann, H. Almstedt, A. Scholz, and M. Oechsner. *Notch Support for LCF-Loading: A Fracture Mechanics Approach*, Procedia Structural Inegrity, Science Direct, Elsevier, Vol. 2, Pages 3125 – 3134, 2016.

[60] M. Hoffmann and T. Seeger. *A Generalized Method for Estimating Multiaxial Elastic-Plastic Notch Stresses and Strains*, part 1: Theory. Journal of Engineering Materials and Technology, 107:250:254, Oct. 1985.

[61] O. Hertel and M. Vormwald. *Statistical and geometrical size effects in notched members based on weakest-link and short-crack modelling*, Engineering Fracture Mechanics, Vol. 95, Pages 72 – 83, Nov. 2012.

[62] T. Beck, H. Gottschalk, L. Mäde, and S. Schmitz. *Combined Notch and Size Effect Modeling in a Local Probabilistic Approach for LCF*, Computational Materials Science, Vol. 142, Pages 377 – 388, Sep. 2017.