



Bergische Universität Wuppertal

Fachbereich Mathematik und Naturwissenschaften

Institute of Mathematical Modelling, Analysis and Computational  
Mathematics (IMACM)

Preprint BUW-IMACM 18/05

H. Gottschalk and M. Saadi

**Shape Gradients for the Failure Probability of a  
Mechanic Component under Cyclic Loading  
A Discrete Adjoint Approach**

June 12, 2018

<http://www.math.uni-wuppertal.de>

# SHAPE GRADIENTS FOR THE FAILURE PROBABILITY OF A MECHANIC COMPONENT UNDER CYCLIC LOADING — A DISCRETE ADJOINT APPROACH —

HANNO GOTTSCHALK and MOHAMED SAADI

Fachgruppe für Mathematik und Informatik, Bergische Universität Wuppertal, Germany

{hanno.gottschalk,saadi}@uni-wuppertal.de

June 12, 2018

## Abstract

This work provides a numerical calculation of shape gradients of failure probabilities for mechanical components using a first discretize, then adjoint approach. While deterministic life prediction models for failure mechanisms are not (shape) differentiable, this changes in the case of probabilistic life prediction. The probabilistic, or reliability based, approach thus opens the way for efficient adjoint methods in the design for mechanical integrity. In this work we propose, implement and validate a method for the numerical calculation of the shape gradients of failure probabilities for the failure mechanism low cycle fatigue (LCF), which applies to polycrystalline metal. Numerical examples range from a bended rod to a complex geometry from a turbo charger in 3D.

**Key words:** Shape Gradients, Failure Probabilities, Adjoint Equation for Structural Mechanics

**Mathematics Subject Classification (2010):** 49Q12, 74P10, 65C50

## 1 Introduction

Mechanical components fail under cyclic loading with stresses well below the ultimate tensile strength (UTS) of the material. This material degeneration process is known as fatigue [7, 31]. In polycrystalline metal, the number of identical load cycles until the initiation of a fatigue crack exposes a statistical scatter which, even under controlled lab conditions, is almost an order of magnitude bigger than, say, the 50% quantile of the data [31]. The physical origin of the degradation process is the gliding of linear lattice defects (displacements) along crystallographic planes of densest packing in the direction of so-called slip systems [21]. Under the cyclic reversal of loads, the displacements form intrusions and extrusions at the surface of the component, from which a crack can start to grow, see [21, 7, 31] and Fig. 1. This is known as low cycle fatigue (LCF) which, in particular, is a surface driven failure mechanism. The random nature of LCF crack

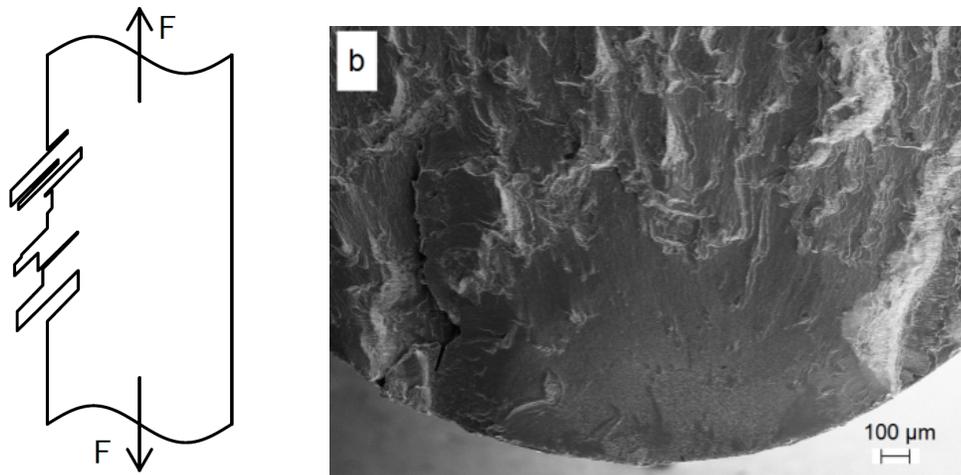


Figure 1: (a) Intrusions and extrusions form on the surface, (b) a crack initiation from the surface (taken from [20]).

formation is then attributed to the random grain structure in a polycrystalline metal. Here, among other effects, the random relative orientation of slip systems relative to the principal stresses and crack percolation over grain boundaries play an important rôle [39].

One of the most frequent design objectives in mechanical engineering is mechanical integrity, which here is understood as the absence of cracks. While the loads are calculated with a finite element (FEA) discretization of the (linear) elasticity partial differential equation (PDE), the save life (or design life), i.e. the number of cycles which the component may be safely used, is then calculated for each point (finite element node) at the surface of the component. Here, the local stress values from the FEA are put into some stress-life or strain-life curves, see e.g. [7, 31] or Section 2. Taking the minimum of all these values – in combination with safety factors which account for the scatter band in the material curves – then results in the safe life of a component. The design objective 'mechanical integrity' can then be understood as the task to maximize the safe life.

Automated optimization of mechanical components has become a standard process in mechanical engineering. Here the wish for an efficient calculation of the sensitivity of design objectives in terms of small changes in the geometry has lead to the implementation of (discrete) adjoint equations in commercially available or open source computational fluid dynamics (CFD) solvers. Besides the usefulness of gradients for optimization purposes, the local sensitivities themselves provide also valuable insight to the designers. Unfortunately, the situation up to now is different for the design objective of 'mechanical integrity'. The reason is properly the deterministic lifing approach, which is based on a (non differential) minimum over a large number of nodes, instead of an integral quantity in the flow variables, as in CFD. This is far more than just a mathematical concern. When this is ignored, as it sometimes happens in the engineering practice, gradient based optimizers tend to jump from one branch of the min-function to the other, if the point of highest loading has a cross-over from one part of the component to some (remote) part of it, so that the convergence process is disrupted.

It has been observed in [19, 10, 34] that the above three problem statements are

interconnected. A proper modeling of the random nature of the LCF crack-initiation process will automatically result in an integral expression in the local stress fields for the acute risk exposure (hazard rate) of the component, which contains the same information as failure probabilities. An adequate model has been proposed in [35, 36, 37, 29], see also [23] and [4] for a related approaches. Therefore, the passage to a probabilistic design prospective with respect to mechanical integrity, as a by-product, solves the problem of differentiability of objective functionals. Analytic calculations using the continuous calculus of shapes can be found in [22, 38], see also [3]. In this work, we report on the first numerical calculation of shape gradients of the aforementioned probabilistic objective functionals, encoding risk exposure for LCF failure. The method of choice is the 'first discretize, then adjoint' approach which can be realized using standard commercial or open source FEA solvers for the adjoint equation [5, 12, 40]. This would not be true for a 'first adjoint, then discretize' approach where derivatives of the original solution up to second order are required, see [38]. These are usually not available in the  $H^1$ -element classes of the commercial solvers. The 'first discretize, then adjoint' approach chosen here therefore seems to us more compatible with industrial application.

Let us underline that it is only the combination of well known ingredients that has lead to this work. Local functionals for the prediction of the UTS of ceramic materials have a history that dates back to Weibull [42, 8]. In the design for ceramic materials, these methods are standard in the science of ceramic materials [17]. Here also the usage of these models in FEA-post-processors with application, e.g., to ceramic heat shields can be found, see e.g. [33]. See also [28] and articles related to the DARWIN tool suite<sup>1</sup> for further probabilistic post processing approaches.

The gradient based, non parametric optimization of shapes, including topology optimization, is a well established and advanced mathematical field, both from the analytical, see [38, 22, 40], and the numerical side, see [2, 12, 5] for incomplete lists. The impressive results obtained here in the case of mechanical design are mostly linked to another design objective, namely the global stiffness of the component, encoded in the compliance as objective functional. This stands for a design intent which is different from mechanical integrity.

Our paper is organized as follows: In Section 2 we review the classical LCF life prediction and the related probabilistic model. Section 3 explains the implementation of this model as a FEA post processor. Here we essentially follow [35, 36, 34]. Section 4 contains the derivation of the shape gradients on the basis of the discrete adjoint equation. Some lengthy calculations have been transferred to Appendix A. Section 5 discusses some detailed information on the implementation and presents two numerical test cases: firstly, a bended rod exposed to a tensile load and secondly a complex turbo geometry (taken from [43]). Finally, in Section 6, we draw our conclusions and give an outlook to future work. Some of our results have been announced in [18].

---

<sup>1</sup>[www.darwin.swri.org/](http://www.darwin.swri.org/)

## 2 Failure Probabilities for Low Cycle Fatigue

### 2.1 The Elasticity PDE

Let  $\Omega \subseteq \mathbb{R}^3$  a bounded region filled with some polycrystalline metal. This represents the shape of the component initially in force free equilibrium. By  $H_D^1(\Omega, \mathbb{R}^3)$  we denote the Sobolev space of  $L^2(\Omega, \mathbb{R}^2)$  functions  $u$  with weak 1st derivatives  $\nabla u \in L^2(\Omega, \mathbb{R}^{3 \times 3})$  such that  $u = 0$  on  $\partial\Omega_D$  [14]. Here  $\partial\Omega$  is a piecewise Lipschitz boundary such that the part with Dirichlet boundary  $\partial\Omega_D$  is an open portion of  $\partial\Omega$  with non vanishing surface measure. Furthermore we set the part of natural boundary conditions to be  $\partial\Omega_N = (\partial\Omega \setminus \partial\Omega_D)^\circ$ , where  $^\circ$  stands for the open interior in  $\partial\Omega$ .

The function  $f \in L^2(\Omega, \mathbb{R}^3)$  stands for a volume force density like imposed by gravity or centrifugal loads.  $g \in L^2(\partial\Omega_N, \mathbb{R}^3)$  is a surface load, like caused by static gas pressure  $P(x)$ ,  $x \in \partial\Omega_N$ ,  $g(x) = -P(x)n(x)$  with  $n(x)$  the outward normal vector field on  $\partial\Omega_N$ . Let  $u \in H^1(\Omega, \mathbb{R}^3)$  be the displacement field caused by these loads in a way that will be specified shortly. We define  $\varepsilon(u) = \frac{1}{2}(\nabla u + \nabla u^T) \in L^2(\Omega, \mathbb{R}^{3 \times 3})$  the linearized strain tensor field and we define the stress tensor field  $\sigma \in L^2(\Omega, \mathbb{R}^{3 \times 3})$  via  $\sigma(u) = \lambda \text{tr}(\varepsilon(u))\mathcal{J} + 2\mu\varepsilon(u)$ , where  $\lambda, \mu > 0$  are the Lamé coefficients that represent the macroscopically isotropic elastic behavior of the material.  $\mathcal{J}$  is the unit matrix on  $\mathbb{R}^3$ ,  $\text{tr}$  is the trace on the space of  $3 \times 3$  matrices and  $M^T$  stands for the transpose of a matrix  $M$ .

It is well known that under the given conditions, the PDE system of linear elasticity in the weak form

$$B(u, v) = \int_{\Omega} f \cdot v \, dx + \int_{\partial\Omega_N} g \cdot v \, dA, \forall v \in H_D^1(\Omega, \mathbb{R}^3) \quad (1)$$

has an unique solution  $u \in H^1(\Omega, \mathbb{R}^3)$  with the bilinear form given by

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

Here  $\cdot$  stands for the scalar product on  $\mathbb{R}^3$  and  $\varepsilon(u) : \varepsilon(v) = \text{tr}(\varepsilon(u)\varepsilon(v))$  is the contraction of both indices of  $\varepsilon(u)$  and  $\varepsilon(v)$ ,  $dx$  is Lebesgue measure and  $dA$  is the surface measure on  $\partial\Omega$ . In fact this follows from Korn's [14] inequality which for the present case gives the coercivity  $A^{-1}\|u\|_1^2 \leq B(u, u) \leq A\|u\|_1^2 \forall u \in H_D^1(\Omega, \mathbb{R}^3)$  for some constant  $\infty > A > 0$  which is needed to apply the Lax-Milgram theorem.

In the following, some operations will frequently require higher degrees of regularity of the solution  $u$  than merely being a  $H^1$ -Sobolev function. In particular this applies to the pointwise evaluation of the stress tensor in deterministic LCF life prediction or the restriction of the stress tensor to the boundary in the local, probabilistic model for LCF life. Both operations can be made rigorous using elliptic regularity theory, see e.g. [1, 19] for suitable conditions on boundary regularity and of the loads  $f$  and  $g$ .

### 2.2 The Equations of Ramberg-Osgood and Coffin-Manson Basquin

In this section we briefly review the deterministic LCF life prediction procedure which forms the basis of the probabilistic model. We closely follow the materials science literature

[7, 21, 31].

For  $x \in \partial\Omega$  let  $\sigma = \sigma(u)$  be the stress tensor from the solution of (1) evaluated at  $x$ . For simplicity we assume that the other extreme of the load cycle is the unloaded situation  $u = 0$ . If this is not the case, we can make use from the linearity of the elasticity PDE and set  $f = f_1 - f_2$  and  $g = g_1 - g_2$ , where  $f_j$  and  $g_j$ ,  $j = 1, 2$ , are the volume and surface loads at the extremes of the load cycle, see Figure 2 (a). As hydrostatic stress can not induce shear stress<sup>2</sup>, we project  $\sigma$  to the trace-free matrices via  $\sigma' = \sigma - \frac{1}{3} \text{tr}(\sigma)\mathcal{J}$ . In the next step we use the Hilbert Schmid norm  $\sigma_v^2 = \frac{3}{2} \sigma' : \sigma' \in \mathbb{R}_+$  to define the elastic von Mises comparison stress. The normalization is chosen in order to reproduce the first principal stress in the case of an uniaxial stress state. Setting  $\sigma_a = \frac{1}{2} \sigma_v \in \mathbb{R}_+$  we arrive at the so called elastic amplitude stress.

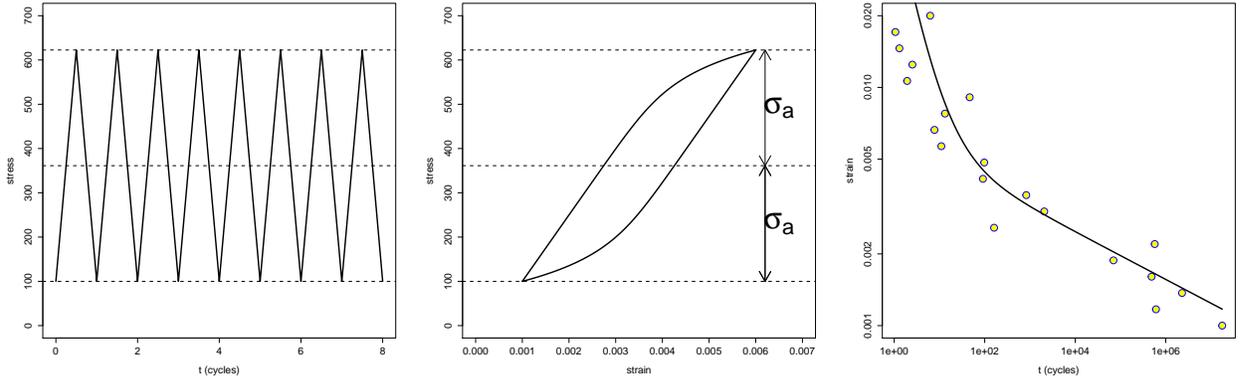


Figure 2: (a) Load cycles, (b) elastic-plastic stress-strain hysteresis with branches according to the Ramberg-Osgood equation, stress in MPa, (c) Coffin-Manson-Basquin curve fitted to specimen test results (data simulated on the basis of realistic values). Note that both axes are on log-scale.

Let us briefly consider elastic-plastic material behavior based on a load cycle with an elastic-plastic amplitude stress  $\sigma_a^{\text{el-pl}}$ . We postpone a mathematical definition for a short moment. Then, the elastic-plastic stress amplitude is connected to the elastic-plastic strain amplitude via the Ramberg-Osgood (RO) relation [32]

$$\varepsilon_a^{\text{el-pl}} = \text{RO}(\sigma_a^{\text{el-pl}}) = \frac{\sigma_a^{\text{el-pl}}}{E} + \left( \frac{\sigma_a^{\text{el-pl}}}{K} \right)^{1/n'}, \quad (3)$$

see also Fig. 2 (b).  $E = \frac{\mu(3\lambda+2\mu)}{\lambda+\mu}$  is Young's modulus,  $K$  is the hardening coefficient and  $n'$  is the hardening exponent.

Finally we use the Coffin-Manson Basquin (CMB) equation to predict the component's LCF-life, i.e. the number of cycles to crack initiation  $N_{\text{det}}$ , via

$$\varepsilon_a^{\text{el-pl}} = \text{CMB}(N_{\text{det}}) = \frac{\sigma_f'}{E} (2N_{\text{det}})^b + \varepsilon_f' (2N_{\text{det}})^c. \quad (4)$$

<sup>2</sup>The slip directions  $s$  mentioned in the introduction are perpendicular to the normal  $\nu$  of the plane of densest packing, thus  $s \cdot \sigma \nu = 0$  if  $\sigma = \sigma_0 \mathcal{J}$ ,  $\sigma_0 \in \mathbb{R}$ , and no force acts on the slip system  $s$  [21].

Here,  $\varepsilon'_f, \sigma'_f > 0$  and  $b < c < 0$  are material constants that are determined from experimental data, see Fig. 2 (c).

Let us now return to the definition of the elastic-plastic stress amplitude  $\sigma_a^{\text{el-pl}}$  on the basis of the stress tensor field  $\sigma = \sigma(u)$  obtained from the solution of (1). In the engineering practice, this is often done with elastic-plastic stress conversion rules by Neuber and Glinka [30, 24]. Let us, for simplicity, explain the Neuber shake-down rule, only. It is obtained by equating the (uniaxial) elastic-plastic energy density  $\varepsilon_a^{\text{el-pl}} \sigma_a^{\text{el-pl}}$  and the corresponding purely elastic quantities  $\sigma_a \varepsilon_a = \frac{\sigma_a^2}{E}$ . As  $\sigma_a$  is calculated on the basis of the elasticity PDE, we can now use the Ramberg-Osgood equation to obtain

$$\sigma_a = \text{SD}(\sigma_a^{\text{el-pl}}) = \sqrt{E \left( \frac{(\sigma_a^{\text{el-pl}})^2}{E} + \sigma_a^{\text{el-pl}} \left( \frac{\sigma_a^{\text{el-pl}}}{K} \right)^{1/n'} \right)}. \quad (5)$$

This gives the following formula for the deterministic life prediction:

$$\text{Ni}_{\text{det}}(\sigma(x)) = \text{CMB}^{-1} \circ \text{RO} \circ \text{SD}^{-1}(\sigma_a(x)), \quad \text{Ni}_{\text{det}} = \inf_{x \in \partial\Omega} \text{Ni}_{\text{det}}(\sigma(x)). \quad (6)$$

We note that there exist various variants of the above general scheme - e.g. one might prefer to apply the shake down to the von Mises stress  $\sigma_v$  instead of the amplitude stress. The numerical differences, especially after fitting the parameters to identical experimental data sets, in most cases are marginal. In real design applications, safety factors are used in various parts of the procedure in order to account for the scatter in the experimental data and modeling errors.

### 2.3 The Local Probabilistic Model for LCF Failure

In the local, probabilistic model for LCF [19, 34, 35, 36, 37] crack formation is modeled as a random event in space, i.e. in  $x \in \partial\Omega$ , and time  $\geq 0$  counting the number load cycles that the component has undergone until the formation of a crack. Although  $t$  strictly speaking is a integer number, already in (4) we tacitly passed to continuous time.

Let thus  $[s, t]$ ,  $0 \leq s < t$ , be a time interval and let  $A \subseteq \partial\Omega$  be some surface region. We then denote by  $\gamma([s, t], A)$  the random number of cracks that initiated in  $A$  during the time interval  $[s, t]$ . In mathematical terms,  $\gamma$  is a so-called point process [25]. If we assume that cracks in non-intersecting regions  $A_1, \dots, A_n$  of  $\partial\Omega$  this initial phase do not interact with each other, i.e.  $\gamma([0, t], A_1), \dots, \gamma([0, t], A_n)$  are independent random variables, we can deduce under quite general assumptions that  $\gamma([0, t], A)$  must be Poisson point process (PPP) [25, 41]. In particular, there exists a sigma finite measure  $\rho(t, \cdot)$  on  $\partial\Omega$  such that

$$P(\gamma([0, t], A) = n) = e^{-\rho(t, A)} \frac{\rho(t, A)^n}{n!}, \quad n \in \mathbb{N}. \quad (7)$$

Here  $P$  is the underlying probability measure. We also note that  $\rho(t, A)$  is the expected number of cracks that originated in  $A$  up to time  $t$ , i.e.  $\mathbb{E}[\gamma([0, t], A)] = \rho(t, A)$ .

Let us shortly review some modeling principles that lead to the model. First of all, the expected number of cracks  $\rho(t, A)$  should only depend on the local stress state  $\sigma(x)$

on  $A$ . Furthermore  $\rho(t, A)$  has to be monotonically increasing in  $t$  as cracks, one they initiated, will stay in place. This leads to

$$\rho(t, A) = \int_0^t \int_A \varrho(t, \sigma(x)) dA dt, \quad (8)$$

where we omitted the possibility that the local crack formation intensity up to time  $t$  could also depend on derivatives of the stress tensor field.

We now need a reasonable model for the local crack formation intensity  $\rho(t, \sigma(x))$ . On the one hand, this model should reflect the experimentally well established procedure of deterministic LCF-life prediction as a kind of average (or quantile) value. On the other hand, the model should allow to adjust itself to different levels of statistical scatter in LCF-life times. We choose the following

$$\varrho(t, \sigma(x)) = \frac{\bar{m}}{\text{Ni}_{\text{det}}(\sigma(x))} \left( \frac{t}{\text{Ni}_{\text{det}}(\sigma(x))} \right)^{\bar{m}-1}, \quad (9)$$

where  $\text{Ni}_{\text{det}}(\sigma(x))$  as defined in (6) is the scale parameter and  $\bar{m} > 0$  is the shape parameter in the sense of the scale-shape class of probability distributions, see e.g. [16] for a discussion with application to reliability statistics. In the limit  $\bar{m} \rightarrow \infty$ , we recover the deterministic model. We then get,

$$\rho(t, A) = \int_A \left( \frac{t}{\text{Ni}_{\text{det}}(\sigma(x))} \right)^{\bar{m}} dA. \quad (10)$$

We define the surface integral type objective functional as

$$J(\Omega, u) = \int_{\partial\Omega} \left( \frac{1}{\text{Ni}_{\text{det}}(\sigma(x))} \right)^{\bar{m}} dA. \quad (11)$$

We now deduce the probability of failure  $\text{PoF}(t)$  – understood as the presence of a LCF crack – as a function of time (load cycle count)  $t$

$$\text{PoF}(t) = 1 - S(t) = 1 - P(\gamma([0, t], \partial\Omega) = 0) = 1 - e^{-t^{\bar{m}} J(\Omega, u)}, \quad t \geq 0. \quad (12)$$

In (12),  $S(t)$  is the probability of survival, which is defined here as the absence of cracks up to time  $t$  on the entire component's surface  $\partial\Omega$ , hence the event  $\gamma([0, t], \partial\Omega) = 0$ . Equation (12) now follows from (7), (10) and (11). We recognize (12) as a Weibull distribution with scale parameter  $\left( \frac{1}{J(\Omega, u)} \right)^{\frac{1}{\bar{m}}}$  and shape parameter  $\bar{m}$ .

Note however that the CMB constants  $\sigma'_f$  and  $\varepsilon'_f$  are not the same as in the deterministic life calculation, see [37] for the details on the calibration of the probabilistic model, see also Subsection 3.2 below. We also note that  $\bar{m} > 1$  is needed for situation when the hazard rate, which is roughly the probability rate that failure will happen in the next moment, given the component survived until that moment, grows with the time  $t$ . See [16] for further details.

### 3 Discretization with Finite Elements

#### 3.1 Discretization of the Probabilistic Model

The numerical approximation of the failure probability  $\text{PoF}(t)$  in (12) requires the discretization of the objective functional (11), which is given as a surface integral. Let  $\mathcal{T}_h$  be a set of finite elements with reference element  $\widehat{K}$  and let  $T_K : \widehat{K} \rightarrow K$  be the standard transformation map from the reference element to  $K \in \mathcal{T}_h$ . See Appendix A.1 and A.2 for further details. Let thus  $\mathcal{N}_h$  be the collection of all faces  $F$  of finite elements  $K = K(F) \in \mathcal{T}_h$  that lie in  $\partial\Omega$ . The total number of such faces is denoted by  $N_{\text{fc}}^F$ . Let  $\widehat{F}$  be the face in the reference element  $\widehat{K}$  such that  $T_K : \widehat{F} \rightarrow F$ . We chose quadrature points  $\hat{\xi}_l^F$  and weights  $\hat{\omega}_l^F$  on  $\widehat{F}$ . We then get from (11), if we insert the approximate solution  $u$  of (27),

$$\begin{aligned} J(\Omega, u) &= \sum_{F \in \mathcal{N}_h} \int_F \left( \frac{1}{\text{Ni}_{\det}(\sigma(x))} \right)^{\bar{m}} dA \\ &= \sum_{F \in \mathcal{N}_h} \int_{\widehat{F}} \left( \frac{1}{\text{Ni}_{\det}(\sigma(T_{K(F)}(\hat{x})))} \right)^{\bar{m}} \sqrt{\det g_F(\hat{x})} d\hat{A} \\ &\approx \sum_{F \in \mathcal{N}_h} \sum_{l=1}^{l_q^F} \hat{\omega}_l^F \left( \frac{1}{\text{Ni}_{\det}(\sigma(T_{K(F)}(\hat{\xi}_l^F)))} \right)^{\bar{m}} \sqrt{\det g_F(\hat{\xi}_l^F)}. \end{aligned} \quad (13)$$

Here  $\det g_F(\hat{\xi})$  is the Gram determinant, i.e. the determinant of

$$g_F(\hat{\xi}) = \widehat{\nabla}^F(T_K \upharpoonright \widehat{F})(\hat{\xi}) \left( \widehat{\nabla}^F(T_K \upharpoonright \widehat{F}) \right)^T(\hat{\xi}), \quad \hat{\xi} \in \widehat{F}, \quad (14)$$

with  $\widehat{\nabla}^F$  the (component wise) gradient on  $\widehat{F}$ . Here  $\sigma(x) = \lambda \nabla \cdot u(x) \mathcal{J} + 2\mu \varepsilon(u(x))$  can be found by expanding  $u$  in the global basis functions  $\theta_j$ , we get for  $u(\xi)$ ,  $u(\xi) = \sum_{j=1}^N u_j \theta_j(\xi) = \sum_{K \in \mathcal{T}_h} \sum_{m=1}^{n_{\text{sh}}} u_{j(K,m)} \widehat{\theta}_m \circ T_K^{-1}(\xi)$  so that

$$\nabla u(\xi) = \sum_{m=1}^{n_{\text{sh}}} u_{j(K,m)} \otimes (\widehat{\nabla} T_K(\hat{\xi})^T)^{-1} \widehat{\nabla} \widehat{\theta}_m(\hat{\xi}), \quad \text{for } \xi \in K \text{ and } \xi = T_K(\hat{\xi}). \quad (15)$$

Symmetrizing the above Jacobi matrix, we obtain  $\varepsilon(u(\xi))$ . Finally, we have

$$\nabla \cdot u(\xi) = \sum_{m=1}^{n_{\text{sh}}} \text{tr} \left( u_{j(K,m)} \otimes (\widehat{\nabla} T_K(\hat{\xi})^T)^{-1} \widehat{\nabla} \widehat{\theta}_m(\hat{\xi}) \right) \quad \text{for } \xi \in K \text{ and } \xi = T_K(\hat{\xi}). \quad (16)$$

Note that  $\widehat{\nabla} T_K(\hat{\xi}) = \sum_{j=1}^{n_{\text{sh}}} \widehat{\nabla} \widehat{\theta}_j(\hat{\xi}) X_{K,j}$  is easily calculated from (25). For the discretized objective functional, we also use the notation  $J(X, U)$ , where  $X$  and  $U$  the  $N \times 3$ -tensor of global coordinates and global degrees of freedom, respectively.

#### 3.2 Numerical Validation and a First Application

As a first application, we present the calculation of crack initiation PoF over the number of load cycles for a radial compressor of a model gas turbine, see [43] for the documentation.

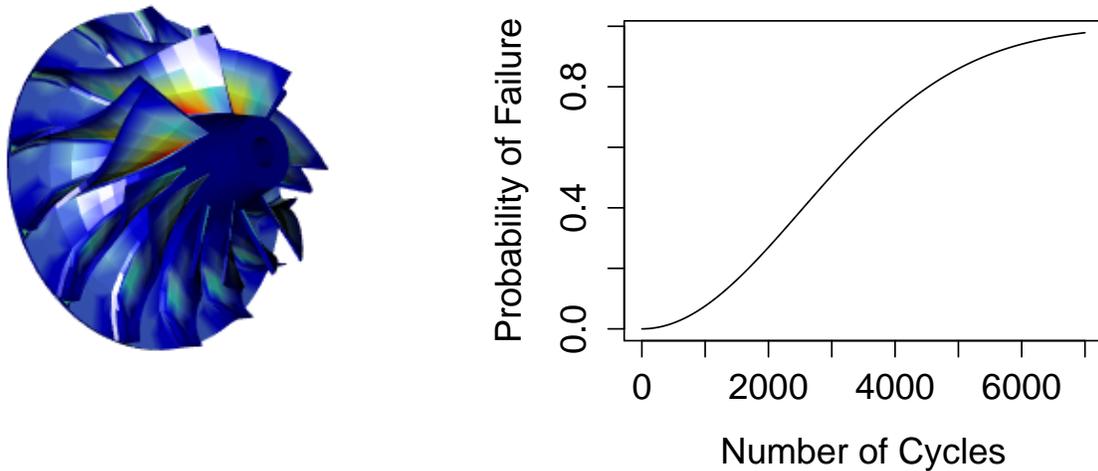


Figure 3: Probability of failure over the number of load cycles (left) for a radial compressor model. (a) crack initiation intensity  $\rho(1, \sigma(x))$  in logarithmic color code from blue (low) to red (high). (b) PoF over the number of load cycles for the compressor model.

The model is part of the CalculiX open source FEA tool suite<sup>3</sup>, which is a design tool for MTU Aero Engines. The outer diameter is 87 mm and the construction material is taken to be the alloy AlMgSi6082, containing roughly 97 wt% Al, roughly 1 wt% Si and Mg along with minor contributions of Mn, Fe and Cr. The detailed chemical composition is described in [11].

Air pressure is neglected, hence only volume forces resulting from the centrifugal load from a rotation speed of 110 000 rpm with a specific density of the material of 2.65 g/cm<sup>3</sup>. The Lamé coefficients for the material are  $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} = 40\,385$  MPa and  $\mu = \frac{E}{2(1+\nu)} = 26\,923$  MPa are calculated from Young’s modulus ( $E = 70\,000$  MPa) and Poisson’s ratio  $\nu = 0.3$ .

The compressor consists of 7 symmetric segments with 2 slightly different blades each, so that the actual FEA model only represents one seventh of the construction with cyclic boundary conditions. Dirichlet boundary conditions are set at the bore. The FEA mesh consists out of 47 971 nodes with 1 302 brick elements of type C3D20R with 20 degrees of freedom and reduced quadrature  $l_q = 8$ . The surface quadrature points are selected such that  $l_q^F = 16$ , confer [35] for a numerical convergence study in the number of quadrature points for a related application.

Ramberg-Osgood Parameters  $n' = 0.064$  and  $K = 443.9$  are reported in [11]. The CMB parameters are given in the top row of Table 1. The passage from deterministic CMB parameters [11] to probabilistic the CMB parameters needed here is performed in

<sup>3</sup>[www.calculix.de](http://www.calculix.de)

	$\sigma'_f$ (MPa)	$\epsilon'_f$	$b$	$c$	$\bar{m}$	$ \partial\Omega $ (mm <sup>2</sup> )
Deterministic ( $q_{0.5}$ -curve)	487	0.209	-0.593	-0.07	-	377
Weibull scale ( $q_{0.63}$ -curve)	436	0.206	-0.593	-0.07	2	377
Probabilistic	2536 [MPa $\times$ mm <sup><math>\frac{2b}{m}</math></sup> ]	0.254 [mm <sup><math>\frac{2c}{m}</math></sup> ]	-0.593	-0.07	2	1

Table 1: Material parameters for the probabilistic model and their deterministic counterparts (taken from [11]). The parameter  $\bar{m}$  is a guess on the basis of probabilistic investigation of other polycrystalline metals.

two steps.

We interpret the reported deterministic values as belonging to a 50% quantile curve. This is justified, as in engineering these curves are mostly obtained as least square fits to the logarithmic specimen data in both dimensions  $\epsilon_a^{\text{le-pl}}$  and  $N_{\text{det}}$ . This fit corresponds to a maximum likelihood fit with log-Normal residuals, where the average curve for the logarithmic data corresponds to a 50%-quantile curve in the non logarithmic data. Within our Weibull model, the relation between the 50% quantile  $q_{0.5}$  and the scale variable  $\eta$  is  $\eta = q_{0.5}/\log(2)^{\frac{1}{\bar{m}}}$ , and we use this to rescale the CMB parameters according with a factor  $\log(2)^{-\frac{b}{\bar{m}}}$  for  $\sigma'_f$  and  $\log(2)^{-\frac{c}{\bar{m}}}$  for  $\epsilon'_f$ , see [37]. The result is given in the second row of Table 1.

In the next step, we account for statistical size effect, i.e. the effect that larger structures have the tendency to fail earlier than smaller ones. In order to convert the CMB parameters that correspond to the scale variable curve of the specimen to the probabilistic CMB parameters of a fictitious specimen with unit surface, we have to apply the scaling relation

$$\frac{\sigma'_f(|\partial\Omega_1|)}{\sigma'_f(|\partial\Omega_2|)} = \left(\frac{|\partial\Omega_1|}{|\partial\Omega_2|}\right)^{\frac{b}{\bar{m}}}, \quad \frac{\epsilon'_f(|\partial\Omega_1|)}{\epsilon'_f(|\partial\Omega_2|)} = \left(\frac{|\partial\Omega_1|}{|\partial\Omega_2|}\right)^{\frac{c}{\bar{m}}}, \quad (17)$$

see [37, 34, 31]. The result is given in the third row of Table 1. The value for  $|\partial\Omega| = 1$  corresponds to the probabilistic parameter in (4). Note that due to (17) this depends on the length scale, which is [mm] in this work (in agreement with CAD models in turbo machinery applications). The value for  $\bar{m}$  for AlMgSi6082 is not reported in the literature. However, experience with steel and Ni-based superalloys shows that  $\bar{m}$  is in the region of 2 for a large range of materials.

The FEA model is solved for  $u$  with the commercial solver ABAQUS 6.1 on a laptop with Intel Core i7-3632QM CPU @ 2.20 GHz with 12GB RAM with a time consumption of 28 seconds. Peak v. Mises stress at the bore is roughly 310 MPa. The surface quadrature for the calculation of  $J(\Omega, u)$  is calculated with an script in the R 3.2.1 and consumed 19 sec execution time on a single core. The obtained  $J$ -value has to be multiplied by 7 to account for the seven rotated sectors.

The resulting total  $J$ -value is  $7.8541 \times 10^{-8}$  [cycles <sup>$-\bar{m}$</sup> ], which corresponds to a Weibull scale variable  $\eta = 3568$  cycles. For a Weibull distribution, this corresponds to the  $1 - \frac{1}{e} \approx$

63% quantile. Figure 3 (a) visualizes the local failure intensity (9) and (b) gives the integrated probability of failure (PoF) over the number of cycles  $t$ . The visual inspection shows that not only the bore region, but also the fillet areas contribute significantly to the total value of  $J$ . The safe life of 2000 cycles reported in the CalculiX tool suite corresponds to a PoF of roughly 25%, according to the local, probabilistic model for LCF, which nicely fits into the over all picture for given design example.

## 4 The Shape Sensitivity for the Probability of Failure via the Adjoint Equation

### 4.1 Straight Forward Calculation of the Shape Sensitivity

We first fix some conventions. The mesh nodes  $X = \{X_1, \dots, X_N\}$  represent the discretized geometry  $\Omega$ . Furthermore we recall that  $U = (u_j)_{j=1, \dots, N}$  stands for the global degrees of freedom and  $u = \sum_{j=1}^N u_j \theta_j(x)$  with  $\theta_j$  the global shape functions. In this section we equivalently write  $J(X, U)$  instead of  $J(\Omega, u)$ . Note that  $U$  is a  $N \times 3$  tensor as well, provided we consider the constraints  $u_j = 0$  for  $X_j \in \partial\Omega_D$  as a part of the discretized equation (27), see also Appendix A.2.

$X$  is a  $N \times 3$  tensor and, by definition, for a tensor of scalar  $Q(X, Z)$ , the partial derivative  $\frac{\partial Q(X, U)}{\partial X}$  is an augmented tensor with two additional slots of dimension  $N$  and 3. Here  $U$  stands for some other tensor with unspecified dimensions (or a scalar, if there are none). If  $U$  is an expression that depends on  $X$ ,  $U = U(X)$ , the total derivative of  $Q(X, U(X))$  with respect to  $X$  is  $\frac{dQ(X, U(X))}{dX} = \frac{\partial Q(X, U(X))}{\partial X} + \frac{\partial Q(X, U(X))}{\partial U} \frac{\partial U(X)}{\partial X}$ . Here the last expression has to be understood in the sense that the additional tensor slots generated by the partial  $U$ -differentiation are contracted with the  $U$ -slots in  $\frac{\partial U(X)}{\partial X}$ .

It is now straight forward to see that the probability of failure  $\text{PoF}(t) = \text{PoF}(t, X, U(X))$ , see (12), at a number of cycles  $t \geq 0$  can be differentiated with respect to the geometry  $X$  as follows

$$\frac{d\text{PoF}(t, X, U(X))}{dX} = t^{\bar{m}} \frac{dJ(X, U(X))}{dX} e^{-t^{\bar{m}} J(X, U(X))}. \quad (18)$$

The above equation is already remarkable in the sense that the direction of the gradient with respect to the geometry does not depend on the number of load cycles  $t > 0$ , which excludes the use of the formalism in design to life activities. We now continue the differentiation and obtain

$$\frac{dJ(X, U(X))}{dX} = \frac{\partial J(X, U(X))}{\partial X} + \frac{\partial J(X, U(X))}{\partial U} \frac{\partial U(X)}{\partial X}, \quad (19)$$

where on both sides we have  $N \times 3$  tensors according to our conventions. Note that in the last term on the right hand side  $U$ -degrees of freedom that correspond to Dirichlet boundary conditions do not depend on  $X$  and thus give zero in  $\frac{\partial U(X)}{\partial X}$ .

In the remainder of this subsection we want to discuss the calculation of  $\frac{\partial J(X, U)}{\partial X}$  and  $\frac{\partial J(X, U)}{\partial U}$ . Inspection of (13) shows that it is easier to deal with  $\frac{\partial J(X, U)}{\partial U}$  as here only  $\sigma$  depends on  $U$ , so we do this first. The following algorithm describes the necessary steps. Detailed calculations can be found in Appendix A.3. Note that here it suffices to calculate

all partial derivatives with respect to degrees of freedom that belong to  $\mathcal{T}_h^F$ , which is the set of 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. For simplicity, we suppress loops over the xyz-indices.

**Data:** FE global node set  $X$ , FE connectivity and surface element tables,  
 FE shape functions  $\hat{\theta}_j$  and gradients  $\widehat{\nabla}\hat{\theta}_j$ ,  
 FE solution  $U = (u_j)$ ,  
 FE surface quadrature points and weights,  
 Elasticity and lifting material constants.

**Result:** A  $(3 \times n_{\text{sh}} \times N_{\text{fc}}^F)$  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
1   initialize a  $3 \times n_{\text{sh}} \times l_q^F$ -tensor  $\frac{\partial J^F}{\partial U} \leftarrow 0$ ;
   for all local degrees of freedom,  $k = 1, \dots, n_{\text{sh}}$  do
2     for all surface quadrature points  $\hat{\xi}_l^F, l = 1, \dots, l_q^F$  do
3       Calculate the derivative of (15) and (16) with respect to  $u_k$  at the
       quadrature point  $\hat{\xi}_l^F$ ;
4       Calculate the derivative of the stress tensor  $\sigma(T_{K(F)}(\hat{\xi}_l^F))$ 
       with respect to  $u_k$ ;
5       Calculate the derivative of  $\sigma_v$  with respect to  $u_k$ ;
       Use (6) as well as (3), (4) and (5) to calculate the  $u_k$ -derivative of
        $\text{Ni}_{\text{det}}(\sigma(T_{K(F)}(\hat{\xi}_l^F)))$ ;
6       Use this to calculate the derivative of  $\left(\frac{1}{\text{Ni}_{\text{det}}(\sigma(T_{K(F)}(\hat{\xi}_l^F)))}\right)^{\bar{m}}$ ;
       Multiply this with the surface quadrature weight  $\hat{\omega}_l^F$  and with
       the Gram determinant  $\sqrt{g_F(\hat{\xi}_l^F)}$ ;
7       Store the result in  $\frac{\partial J^F}{\partial U}[:, k, l]$ ;
     end
   end
8   Sum up the  $(3 \times n_{\text{sh}} \times l_q^F)$  tensor  $\frac{\partial J^F}{\partial U}$  obtained over the quadratures;
9   Store the result in  $\frac{\partial J}{\partial U_{\text{loc}}}[:, \cdot, F]$ ;
end

```

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

The previous algorithm provides a surface face-wise separate computation of the partial derivative of the cost functional  $J$  with respect to the local vector of displacements and thus gives a  $(3 \times n_{\text{sh}} \times N_{\text{fc}}^F)$  tensor. To evaluate the global partial derivative, we have to assemble the tensor  $\frac{\partial J}{\partial U_{\text{loc}}}$  based on the surface element and connectivity lists. The following algorithm describes the assembling process.

**Data:** A  $3 \times n_{\text{sh}} \times N_{\text{fc}}^F$  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  $F \in \mathcal{N}_h$  do
  for all local degrees of freedom  $k = 1, \dots, n_{\text{sh}}$  do
1   |  $\frac{\partial J}{\partial U}[\cdot, \hat{j}(K(F), k)] \leftarrow \frac{\partial J}{\partial U}[\cdot, \hat{j}(K(F), k)] + \frac{\partial J}{\partial U_{\text{loc}}}[\cdot, k, F]$ ;
   | end
  end
end

```

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

The algorithm to calculate  $\frac{dJ}{dX}$  is similar to the Algorithm 1, however we have to take the potential dependency of  $\sqrt{g_F(\hat{\xi}_l^F)}$  and  $(\widehat{\nabla}T_{K(F)}(\hat{\xi}_l))^{-1}$  of the  $j$ -th node  $X_j$  into account. The following algorithm highlights the changes that are needed, detailed calculations are given in Appendix A.4. Note that certain steps, as e.g. step 4 in the following algorithm, can be done beforehand, as the result will only depend on element type and quadrature points. We leave such steps in the loop for better readability.

**Data:** Same as in Algorithm 1

**Result:** A  $(3 \times n_{\text{sh}} \times N_{\text{fc}}^F)$  tensor containing  $\frac{\partial J}{\partial X_{\text{loc}}}$

```

1 initialization  $\frac{\partial J}{\partial X_{\text{loc}}} \leftarrow 0$ ;
for all faces  $F \in \mathcal{N}_h$  do
2   | initialize a  $3 \times n_{\text{sh}} \times l_q^F$ -tensor  $\frac{\partial J^F}{\partial X} \leftarrow 0$ ;
   | for all local degrees of freedom,  $k = 1, \dots, n_{\text{sh}}$  do
   |   | for all surface quadrature points  $\hat{\xi}_l^F, l = 1, \dots, l_q$  do
   |   |   | 3 Calculate the derivative of Jacobian matrix  $\widehat{\nabla}T_{K(F)}(\hat{\xi}_l^F)$ 
   |   |   | with respect to  $X_k^{K(F)}$ ;
   |   |   | 4 Calculate the derivative of (15) and (16) with respect to  $X_k^{K(F)}$  using (25);
   |   |   | 5 Calculate the  $X_k$  derivative of  $\sqrt{\det g_F(\hat{\xi}_l^F)}$  using (14) and (25);
   |   |   | 6 Multiply the result with  $\left(\frac{1}{N_{\text{idet}}}\right)^{\bar{m}}$  and  $\widehat{\omega}_{lF}$ ;
   |   |   | 7 Store the result in  $\frac{\partial J^F}{\partial X}[\cdot, k, l]$ ;
   |   |   | 8 Follow the steps 2–6 of Algorithm 1 analogously with  $u_k$  replaced by  $X_k$ ;
   |   |   | 9 Augment  $\frac{\partial J^F}{\partial X}[\cdot, k, l]$  by the result;
   |   |   | end
   |   | end
   | end
10  | Sum up the  $(3 \times n_{\text{sh}} \times l_q^F)$  tensor  $\frac{\partial J^F}{\partial X}$  obtained over the quadratures;
11  | Store the result in  $\frac{\partial J}{\partial X_{\text{loc}}}[\cdot, \cdot, F]$ ;
end

```

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

As described in the Algorithm 2, we have now to assemble  $\frac{\partial J}{\partial X_{\text{loc}}}$  to obtain  $\frac{\partial J}{\partial X}$ .

A numerical calculation of  $\frac{\partial U(X)}{\partial X}$  e.g. by finite differences is possible in principle, but in many instances is not practically feasible as the dimension of  $X$  matches the degrees of freedom, which can be several hundreds of thousands in industrial application. As usually, we avoid the problem of solving a finite element simulation for each of these degrees of freedom by using the Lagrangian formalism in the next subsection.

## 4.2 Shape Sensitivity via the Adjoint Equation

We define the Lagrangian [40] of the discretized problem as

$$\mathcal{L}(X, U, \Lambda) = J(X, U) - \Lambda^T (B(X)U - F(X)). \quad (20)$$

Here, the adjoint state  $\Lambda = (\lambda_j)_{j \in \{1, \dots, N\}}$  is a  $N \times 3$  tensor and the expression  $\Lambda^T(\dots)$  is understood in the sense that the  $(j, r)$  indices in (28) are contracted with the corresponding indices of  $\Lambda$ . If Dirichlet boundary conditions hold for  $U = (u_j)$ , we have the same boundary conditions  $\lambda_j = 0$  for  $X_j \in \partial\Omega_D$  for the adjoint state. As usually,  $\frac{\partial \mathcal{L}(X, U, \Lambda)}{\partial \Lambda} = 0$  gives the state equation (28) and

$$0 = \frac{\partial \mathcal{L}(X, U, \Lambda)}{\partial U} = \frac{\partial J(X, U)}{\partial U} - \Lambda^T B(X) \Leftrightarrow B(X)\Lambda = \frac{\partial J(X, U)}{\partial U} \quad (21)$$

is the adjoint equation. Here we used the symmetry of the stiffness matrix in the  $(j, r)$  and  $(k, l)$  indices, see (28). Assuming that the state equation and the adjoint equation hold, the partial derivative of  $\mathcal{L}(X, U, \Lambda)$  with respect to  $X$  equals the total derivative of  $J(X, U(X))$ , hence

$$\frac{dJ(X, U(X))}{dX} = \frac{\partial \mathcal{L}(X, U, \Lambda)}{\partial X} = \frac{\partial J(X, U)}{\partial X} - \Lambda^T \left( \frac{\partial B(X)}{\partial X} U - \frac{\partial F(X)}{\partial X} \right). \quad (22)$$

The above equation is understood in the sense that all tensor indices from  $\frac{\partial B(X)}{\partial X}$  and  $\frac{\partial F(X)}{\partial X}$  are contracted with  $\Lambda$  and  $U$ , except those that originate from the partial differentiation with respect to  $X$ . Note that the tensor  $\frac{\partial B(X)}{\partial X}$  is of dimension  $(N \times 3) \times (N \times 3) \times (N \times 3)$  and  $\frac{\partial F(X)}{\partial X}$  is still of dimension  $(N \times 3) \times (N \times 3)$ . A one-to-one storage of these quantities in the main memory for real world engineering FE models will be impossible on almost all computer architectures and remain very memory consuming even if sparse data structures are used. We therefore calculate the quantities  $\Lambda^T \frac{\partial B}{\partial X} U$  and  $\Lambda^T \frac{\partial F}{\partial X}$  directly. The following algorithm gives the numerical calculation of the 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 (21) 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 (22).

**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  $\frac{dJ}{dX}$ . The detailed calculations can be found in Appendix A.5 and Appendix A.6. Note that in many cases, it will not be possible to store the  $3 \times n_{\text{sh}} \times 3 \times n_{\text{sh}} \times 3 \times n_{\text{sh}} \times N_{\text{el}}$  tensor  $\frac{\partial B}{\partial X_{\text{loc}}}$  due to constraints of the main memory, even if a sparse data format is used. We circumvent this obstacle by contracting with the original solution  $U$  and the adjoint state  $\Lambda$  during the local calculation. Note that again, for better readability, we did not try to avoid obvious redundant calculations in the loops.

**Data:** Same as in Algorithm 4, but with volume quadrature points and weights.  
The adjoint state  $\Lambda$ .

**Result:** A  $3 \times n_{\text{sh}} \times N_{\text{el}}$  tensor containing  $\Lambda^T \frac{\partial B}{\partial X_{\text{loc}}} U$

```

1 initialization  $\Lambda^T \frac{\partial B}{\partial X_{\text{loc}}} U \leftarrow 0$ ;
  for all elements  $K \in \mathcal{T}_h$  do
2     initialize a  $3 \times n_{\text{sh}} \times 3 \times n_{\text{sh}} \times 3 \times n_{\text{sh}} \times l_q$  tensor  $\frac{\partial B}{\partial X_{\text{loc}}}^K \leftarrow 0$ ;
3     initialize a  $3 \times n_{\text{sh}}$  tensor  $\Lambda_{\text{loc}}^K \leftarrow 0$ ;
4     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
5             Compute  $\frac{\partial}{\partial X_j}(\omega_{lK})$  using (56) ff.;
6             Compute  $\frac{\partial}{\partial X_j}(\nabla_{[\cdot]} \theta_q(\xi_l))$  and  $\frac{\partial}{\partial X_j}(\nabla_{[\cdot]} \theta_k(\xi_l))$  using (58);
7             Compute  $\frac{\partial B}{\partial X_{\text{loc}}}^K[\cdot, j, \cdot, k, \cdot, q, l]$  using (55) and (59);
8             Assign  $\Lambda_{\text{loc}}^K[\cdot, k] \leftarrow \Lambda[\cdot, \hat{j}(K, k)]$ ;
9             Assign  $U_{\text{loc}}^K[\cdot, q] \leftarrow U[\cdot, \hat{j}(K, q)]$ ;
           end
         end
       end
     end
10    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  $l$ ;
11    Store the result in  $\Lambda^T \frac{\partial B}{\partial X_{\text{loc}}} U[\cdot, j, K]$ ;
  end
end
```

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

The assembly of  $\Lambda^T \frac{\partial B}{\partial X_{\text{loc}}} U$  to  $\Lambda^T \frac{\partial B}{\partial X} U$  is similar to Algorithm 2 and is omitted here. Apparently, Algorithm 5 will be very run time intensive. However, it can easily be speeded up by embarrassing parallelism over the elements  $K \in \mathcal{T}_h$ .

The following algorithm gives the derivative of the volume loads with respect to  $X$  for the case where the volume force density does not change with  $X$ . 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 5.

The calculation of the partial derivatives of the volume force term leads to similar storage problems as in Algorithm 5. Again we solve this by an element wise calculation and contraction with the adjoint state  $\Lambda$ . Note that the storage problem is less severe by a  $3 \times N$  factor for the global expressions. Hence an alternative strategy would be a direct storage of the global data in a sparse matrix format. The force vector might (centrifugal load) or might not (gravity) be  $X$ -dependent, see the following section.

**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  $3 \times n_{\text{sh}} \times N_{\text{el}}$  tensor  $\Lambda^T \frac{\partial F^{\text{vol}}}{\partial X_{\text{loc}}}$

```

1 initialization  $\Lambda^T \frac{\partial F^{\text{vol}}}{\partial X_{\text{loc}}} \leftarrow 0$ ;
  for all elements  $K \in \mathcal{T}_h$  do
2     initialize a  $3 \times n_{\text{sh}} \times 3 \times n_{\text{sh}} \times l_q$  tensor  $\frac{\partial F^{K \text{ vol}}}{\partial X_{\text{loc}}} \leftarrow 0$ ;
3     initialize a  $3 \times n_{\text{sh}}$  tensor  $\Lambda_{\text{loc}}^K \leftarrow 0$ ;
     for all local degrees of freedom,  $j = 1, \dots, n_{\text{sh}}$  do
4         for all local degrees of freedom,  $k = 1, \dots, n_{\text{sh}}$  do
5             for all volume quadrature points  $\hat{\xi}_l, l = 1, \dots, l_q$  do
6                 Compute  $\frac{\partial}{\partial X_j}(\omega_{lK})$  using (56) ff.;
7                 Compute  $\frac{\partial f_{[l]}}{\partial X_j}(\xi_l)$  (model specific);
8                 Compute  $\frac{\partial F^{K \text{ vol}}}{\partial X_{\text{loc}}}[\cdot, j, \cdot, k, l]$  using (61) ;
9                 Assign  $\Lambda_{\text{loc}}^K[\cdot, k] \leftarrow \Lambda[\cdot, \hat{j}(K, k)]$ ;
             end
         end
     end
     Multiply  $\frac{\partial F^{K \text{ vol}}}{\partial X_{\text{loc}}}[\cdot, j, \cdot, k, \cdot, q, l]$  with  $\Lambda_{\text{loc}}^K[\cdot, k]$  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  $\Lambda^T \frac{\partial F^{\text{vol}}}{\partial X_{\text{loc}}}$

Also in the case of the  $X$ -derivative of the surface loads one might adjust the force densities, e.g. in order to keep a total force constant when a surface varies. Again, we refer to Section 5 for further discussion.

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

```

1 initialization  $\Lambda^T \frac{\partial F^{\text{surf}}}{\partial X_{\text{loc}}} \leftarrow 0$ ;
  for all faces  $F \in \mathcal{N}_h$  do
2   initialize a  $3 \times n_{\text{sh}} \times 3 \times n_{\text{sh}} \times l_q^F$  tensor  $\frac{\partial F^{K \text{ surf}}}{\partial X_{\text{loc}}} \leftarrow 0$ ;
3   initialize a  $3 \times n_{\text{sh}}$  tensor  $\Lambda_{\text{loc}}^K \leftarrow 0$ ;
     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 surface quadrature points  $\hat{\xi}_l^F$ ,  $l = 1, \dots, l_q^F$  do
4           Compute  $\frac{\partial}{\partial X_j^{K(F)}}(\omega_{lF})$  using (44) ff.;
5           Compute  $\frac{\partial g_{[ \cdot ]}}{\partial X_j^{K(F)}}(\xi_l)$  (model specific);
6           Compute  $\frac{\partial F^{K \text{ surf}}}{\partial X_{\text{loc}}}[\cdot, j, \cdot, k, l]$  using (62) ;
7           Assign  $\Lambda_{\text{loc}}^K[\cdot, k] \leftarrow \Lambda[\cdot, \hat{j}(K(F), k)]$ ;
         end
       end
     end
8   Multiply  $\frac{\partial F^{K \text{ surf}}}{\partial X_{\text{loc}}}[\cdot, j, \cdot, k, \cdot, q, l]$  with  $\Lambda_{\text{loc}}^K[\cdot, k]$  and sum over  $k$  (along with
     the related  $xyz$  index) and quadrature index  $l$ ;
9   Store the result in  $\Lambda^T \frac{\partial F^{\text{surf}}}{\partial X_{\text{loc}}}[\cdot, j, F]$ ;
  end
end
```

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

The assembly to the global  $3 \times N$  tensors  $\Lambda^T \frac{F^{\text{vol}}}{\partial X}$  and  $\Lambda^T \frac{F^{\text{vol}}}{\partial X}$  again is standard. We have thus provided all the necessary sub-algorithms to Algorithm 4, which allows us an efficient calculation of the shape sensitivity.

## 5 Numerical Examples and Validation

### 5.1 Implementation Details

The implementation of the adjoint method is based on scripts in R 3.2.1 and python as well as the commercial solver ABAQUS 6.12. R scripts for the implementation of Algorithm 5 have been parallelized on up to 12 cores using the `parallel` package. Local and some of the global tensor calculations are performed on compiled code using tensor summations and contractions provided by the R package `tensor` 1.5.

Element types available are tetrahedral elements with 4 and 8 DoF and brick elements with 8 or 20 DoF. Quadratures for the surface integrals can be chosen with up to 36 quadrature points, while exact and reduced quadratures are available for volume integrals

Clamped Dirichlet and cyclic boundary conditions have been implemented along with

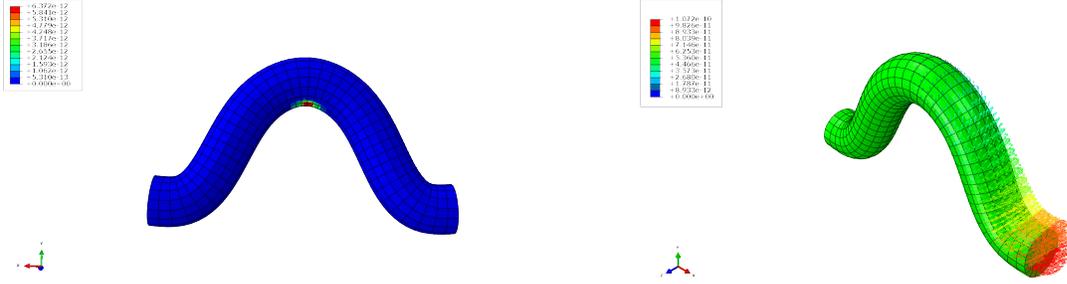


Figure 4: Crack formation intensity and adjoint state: (a) Crack formation intensity for the bended rod. Stress concentration on the lower side of the location with the strongest curvature leads to an augmented probability of stress initiation (left). (b) Adjoint state  $\Lambda_j$  is visualized as an arrow at  $X_j$ . Asymmetry is a consequence of non symmetric boundary conditions.

volume forces and surface forces.

## 5.2 A Bended Rod under Tensile Loading

As a first example we consider a situation where one has an intuitive idea where the shape gradient should point to. We thus consider a geometry of a rod which is 6mm long and is bended up to a height of 3mm. Material properties are the same as in Section 3.2. The diameter is 1mm. The boundary condition is Dirichlet  $u = 0$  at the left and a uniform tensile stress on the right end with 12 N/mm, which makes up a total force of 18.85N over the total surface of 1.57mm<sup>2</sup>. The FE model consists out of 6410 nodes and 1302 brick elements of type C3D20R with  $n_{\text{sh}} = 20$  local degrees of freedom. The volume quadrature contains  $l_q = 8$  points and the surface quadrature contains  $l_q^F = 36$  quadrature points. The resulting  $J$ -value is  $J = 4.20160 \times 10^{-11}$  [cycles<sup>-m</sup>] which corresponds to a Weibull scale variable of  $\eta = 154274$  cycles.

In the calculation of the partial derivative  $\frac{\partial F(X)}{\partial X}$  in (22), we have to take into account the potential  $X$  dependence of the physical force density  $f$ . In the model considered here,  $A$  stands for the portion of the surface where the constant force density is applied in  $e_1$  direction. We adjust the force density  $f$  in the way that  $f_1(X)A(X)$  is kept constant to the value of 18.85 N, where  $A(X)$  is the area of the surface to which the force is applied after a potential deformation due to a change of  $X$ . If this is neglected, the shape gradient would point to the inward direction at  $A$  to decrease the surface and thereby lower to total load. In order to account for effect, notify that  $A(X) = \sum_{F \in \mathcal{N}_h, F \subseteq A} \sum_{l=1}^{l_q^F} \omega_{lF}$  and thus an extra term

$$\frac{\partial f_1(\xi_l^F)}{\partial X_j} = -\frac{f_1}{A(X)^2} \sum_{F \in \mathcal{N}_h, K(F) \in \hat{j}^{-1}(j)_1, F \subseteq A} \sum_{l=1}^{l_q^F} \frac{\partial}{\partial X_j} \omega_{lF}, \quad (23)$$

which has to be inserted into the middle term (62) in Appendix A.6.  $\frac{\partial}{\partial X_j} \omega_{lF}$  is calculated in (44). Here  $\hat{j}^{-1}(j)_1$  is the projection of  $\hat{j}^{-1}(j) = \{(K_1, n_1), \dots, (K_k, n_k)\}$  to the first component  $\{K_1, \dots, K_k\}$ .

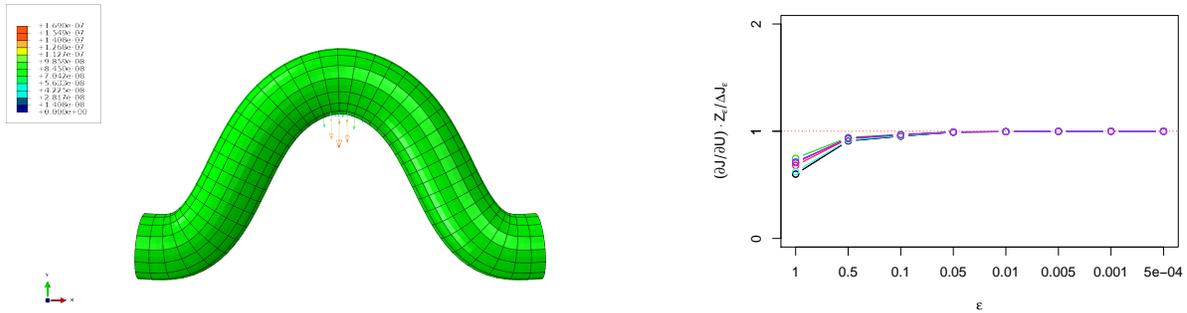


Figure 5: Partial  $U$  derivatives of  $J$ : (a) Partial derivative  $\frac{\partial J}{\partial U_j}$  visualized as an arrow at  $X_j$  (left). (b) Comparison of  $\frac{\partial J}{\partial U} \cdot V$  with finite the differences  $\frac{J(X,U+\varepsilon V)-J(X,U)}{\varepsilon}$  for  $\varepsilon \rightarrow 0$ .  $V$  is taken to be proportional to  $U$  (right).

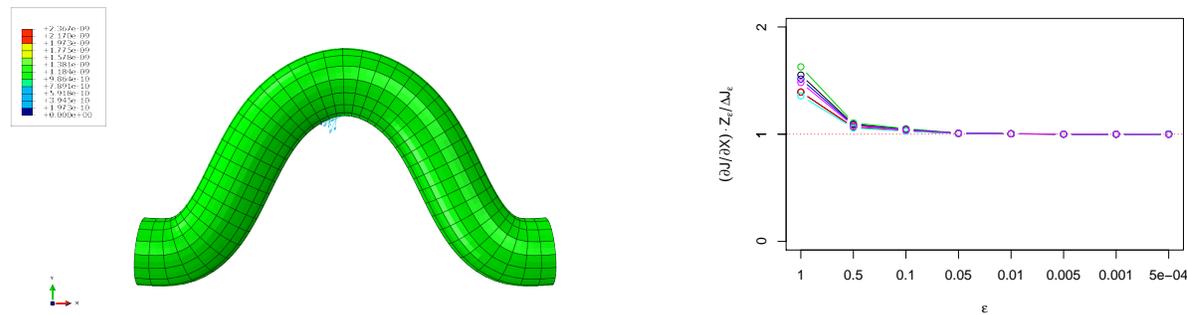


Figure 6: Partial  $X$  derivatives of  $J$ : (a) The  $j$ th component of  $\frac{\partial J}{\partial X}$  is visualized as an arrow at  $X_j$  (left). (b) Comparison of  $\frac{\partial J}{\partial X} \cdot V$  with finite the differences  $\frac{J(X+\varepsilon V,U)-J(X,U)}{\varepsilon}$  for  $\varepsilon \rightarrow 0$ .  $V$  is the normal vector field (right).

Figure 4 (a) displays the crack formation intensity for the bended rod, whereas Figure 4 (b) the adjoint state is displayed. The hot spot for crack initiation is located at the lower side of the portion of the rod with the strongest bending.

Note that for the adjoint state, Dirichlet conditions are imposed on the left hand side, too, however no such conditions hold on the right. This explains the non symmetric appearance, despite  $\frac{\partial J}{\partial U}$  is rather symmetric, see Figure 5 (a).

Figures 5 (a) and 6 (a) display the partial derivative  $\frac{\partial J}{\partial U}$  and  $\frac{\partial J}{\partial X}$ , respectively. The right panels Fig. 5 (a) and Fig. 6 (b) compare results of the the finite difference method with the partial derivatives. The numerical validation reveals relative errors in the range of 0.1%.

The components of the derivative  $\frac{\partial J}{\partial X_j}$  that are parallel to the surface contour at  $X_j$  look artificial. However, such arrows that are pointing away from the location of highest stress can be understood by the interplay of finite element discretization and the probabilistic model. As the function  $\left(\frac{1}{N_{\text{idet}}}\right)^m$  is convex in the von Mises stress, the numerics produces a lower integral result for  $J$  if an element with high loading is stretched, as this decreases stress values, even if this quenches a neighboring element with potentially

lower stress levels. It should however be clear that this is only a by-product of the finite element discretization and not a physical effect. It therefore seems to be reasonable, only to consider normal components of (partial) shape derivatives  $\frac{\partial J}{\partial X}$  and  $\frac{dJ}{dX}$ .

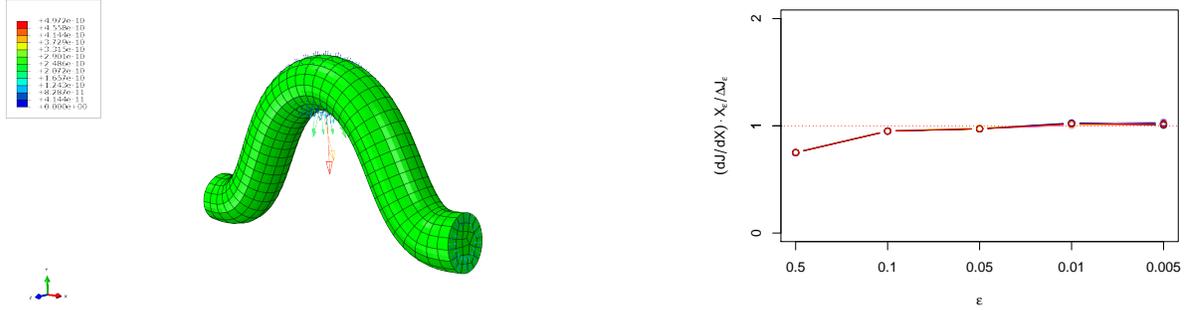


Figure 7: Validation of the shape sensitivities: (a) The  $j$ th component of  $\frac{\partial J}{\partial X}$  is visualized as an arrow at  $X_j$  (left). (b) Comparison of the shape sensitivity  $\frac{dJ}{dX} \cdot V$  with finite the differences  $\frac{J(X+\varepsilon V, U(X+\varepsilon V)) - J(X, U(X))}{\varepsilon}$  for  $\varepsilon \rightarrow 0$ .  $V$  is the vector field proportional to  $X$  (right).

The shape sensitivity is displayed in Figure 7. Panel (a) displays the total shape sensitivity and (b) shows the validation with finite differences in the direction of the node coordinates  $X$  which corresponds to a scaling of the structure.

### 5.3 Shape Sensitivity of the Radial Compressor

We next calculate shape sensitivities in the normal direction in the case of the radial compressor model introduced in Section 3.2. As gas pressure is neglected in this model, the load vector is exclusively determined by the centrifugal load at the rotation speed of 110000 rpm which corresponds to  $\omega = 2\pi \times 1833.33 = 11519.17$  Hz and density  $\rho = 2650$  kg/m<sup>3</sup>. Let  $x$  be the rotation axis and  $\xi^\perp = (0, \xi_2, \xi_3)$ . Then, the centrifugal force density is  $f = \rho\omega^2\xi^\perp$  and thus we have to introduce a term that represents the change of the centrifugal load under the change of node coordinates in  $\frac{\partial F}{\partial X}$ , see Algorithm 6 and (61), as follows

$$\frac{\partial f}{\partial X_{ji}}(\xi_l^K) = \rho\omega^2 \frac{\partial}{\partial X_{ji}} T_K(\hat{\xi}_l)^\perp = \begin{cases} 0 & \text{if } i = 1 \\ \rho\omega^2 \hat{\theta}_j(\hat{\theta}_l) e_i & \text{if } i = 2, 3 \end{cases}, \quad (24)$$

where  $X_{ji}$  is the local node coordinate and we also used (25).

The shape sensitivity for the radial compressor is displayed in figure 8. Also in the case of a complex geometry the shape sensitivity calculations conform to more than 99% with the finite difference method.

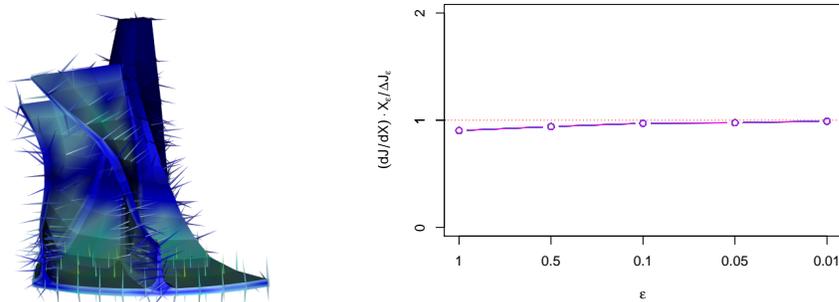


Figure 8: Shape sensitivity for the radial compressor: (a) Shape sensitivity for the radial compressor. Length of  $\frac{dJ}{dX_j}$  is displayed as arrow at  $X_j$  (left). (b) Validation of the compressor sensitivities vs finite differences.

## 6 Conclusion and Outlook

In this work we demonstrated that the probability for Low Cycle Fatigue (LCF) crack initiation is an objective functional for mechanical engineering that permits the calculation of shape sensitivities via the discrete adjoint method. We revised the modeling and numerical implementation of probabilistic LCF life models and described the calculation of shape sensitivities for the probability of failure both from the algorithmic and the analytic standpoint. We also provided numerical validation with the relative total error for the shape sensitivities in the range of a few percent, when compared with finite differences. To the best of our knowledge, this is the first numerical implementation of a shape gradient for failure probabilities. This paves the way for an optimal design for reliability using highly efficient gradient based methods, see [9] for some first steps in 2D.

We discussed some issues of the discrete adjoint method related to the enlarging of highly loaded element. This leads to considerable non normal components in the shape derivative. Projection to the normal direction however is an adequate solution.

It has been demonstrated that the formalism can be implemented with the aid of existing FE solvers using usual desktop architectures with a moderate degree of parallelism. Run times remain moderate even for real world 3D design applications. Nevertheless, run times can possibly considerably reduced in future implementations, as compared with our R based ‘rapid prototype’.

Several directions of future research open up from this point: From the applied standpoint, the extension of the probabilistic approach to further material classes and damage mechanisms is desirable. Also, having gas turbine design in mind, the treatment of the thermo-mechanical system of partial differential equations is desirable. Furthermore, contact boundary conditions at some point should be integrated into the formalism and the implementation.

It would also be interesting to use the probabilistic objective functional along with the adjoint equations for adaptive mesh refinement as proposed in [5].

From a more theoretical standpoint, a better numerical control of the probabilistic objective functional is desirable. Note that this is not straight forward in the present weak and  $H^1$ -based approach, as no trace on  $\partial\Omega$  exists for the stress. Nevertheless, we

suggest that with the aid of elliptic regularity theory [1, 14, 19, 34] and using derivative recovery schemes, see e.g. [6], this problem can probably be resolved.

It also would be desirable to repeat the calculation and implementation of shape derivatives with a first adjoint and then discretize approach and benchmark both strategies for applications in design for optimal reliability. See e.g. [10] for some first analytic steps in this direction. This however will not be possible with standard FE solvers and therefore is probably less close to industrial application. A successful benchmark for this strategy could however boost further research and development in that direction with designs with optimal reliability in view.

Last, it is very important to address the challenge of multi objective optimization in a gradient based context. In gas turbine design, shape sensitivities for the aerodynamic performance is already implemented in many publicly available codes. It is therefore highly desirable to put the mechanical design on an equal footing and use the reduction in algorithmic complexity by the adjoint method in multi-physics optimization.

**Acknowledgements:** We thank the Siemens Gas Turbine Engineering Department for Probabilistic Design for valuable discussions and support, in particular thanks go to Dr. Georg Rollmann and Dr. Sebastian Schmitz. We also thank Dr. T. Seibel (Forschungszentrum Jülich) and Prof. Dr. T. Beck (Technical University of Kaiserslautern) and R. Krause (ICS Lugano) for prior joint work on probabilistic LCF, on which this paper is based. This work has been made possible by financial support under the AG Turbo Grant 4.1.13 funded by Siemens Power and Gas, the German Federal Ministry of Economic Affairs (BMWi) under the grant no. 03ET7041J and by the BMBF project GIVEN under the grant no. 05M2018.

## A Detailed Calculations

### A.1 Notation for Finite Elements

To fix the notation for later purposes, we introduce the standard discretization of the PDE (1) with Lagrange finite elements [15, 13].

Let the finite element be given by  $\{K, P(K), \Sigma(K)\}$  where  $K \subseteq \mathbb{R}^3$  is a compact, connected Lipschitz set,  $P$  is a finite vector space polynomials  $p : K \rightarrow \mathbb{R}$ . The local degrees of freedom are linear functionals on  $P(K)$  defined by  $\varphi_{K,j}(p) = p(X_j)$  where there are nodes  $X_1^K, \dots, X_{n_{\text{sh}}}^K \in K$  such that the mapping from  $P(K)$  to its local degrees of freedom is bijective. The dual basis of  $P(K)$  with respect to the local degrees of freedom are the local shape functions  $\theta_{K,k} \in P(K)$  and  $\varphi_{K,j}(\theta_{K,k}) = \delta_{j,k}$ ,  $j, k \in \{1, \dots, n_{\text{sh}}\}$ .

Let  $\mathcal{T}_h$  be a finite element mesh on  $\Omega$  with  $N_{\text{el}}$  elements. As usually we assume that there is a reference element  $\{\hat{K}, \hat{P}, \hat{\Sigma}\}$  with a finite dimensional linear space of reference polynomials  $\hat{P}$ . For each element  $K \in \mathcal{T}_h$  in the mesh, we assume that there is a bijective transformation  $T_K : \hat{K} \rightarrow K$  such that  $\hat{P} = P \circ T_K$ ,  $\hat{\theta}_j = \theta_j \circ T_K$  and  $\hat{\varphi}_j(p \circ T_K) = \varphi_j(p)$ ,  $j \in \{1, \dots, n_{\text{sh}}\}$ . In many cases, we have

$$T_K(\hat{\xi}) = T_K(\hat{\xi}, X) = \sum_{j=1}^{n_{\text{sh}}} \hat{\theta}_j(\hat{\xi}) X_j^K, \quad \hat{\xi} \in \hat{K}. \quad (25)$$

Let  $X = \{X_1, \dots, X_N\} = \bigcup_{K \in \mathcal{T}_h} \{X_1^K, \dots, X_{n_{\text{sh}}}^K\}$  be the set of all the Lagrange nodes. For  $K \in \mathcal{T}_h$  and  $m \in \{1, \dots, n_{\text{sh}}\}$ , let  $\hat{j}(K, m) \in \{1, \dots, N\}$  be the corresponding index of the Lagrange node. The mapping  $\hat{j}(\cdot, \cdot) : \mathcal{T}_h \times \{1, \dots, n_{\text{sh}}\} \rightarrow \{1, \dots, N\}$  is also called the connectivity. Let  $\{\theta_j : j \in \{1, \dots, N\}\}$ , be the set of global shape functions  $\theta_j : \Omega \rightarrow \mathbb{R}$ . When these functions are restricted to  $K \in \mathcal{T}_h$ , they belong to  $P(K)$  and fulfill

$$\theta_j \upharpoonright_K = \hat{\theta}_k \circ T_K^{-1} \text{ provided } \exists k \in \{1, \dots, n_{\text{sh}}\} \text{ such that } \hat{j}(K, k) = j. \quad (26)$$

The global finite element space  $H_h^1(\Omega, \mathbb{R})$  is the linear span of  $\{\theta_j : j \in \{1, \dots, N\}\}$ . Let  $H_h^1(\Omega, \mathbb{R}^3) = H_h^1(\Omega, \mathbb{R})^{\times 3}$ . We have  $H_h^1(\Omega, \mathbb{R}^3) \subseteq H^1(\Omega, \mathbb{R}^3)$  and let  $H_{D,h}^1(\Omega, \mathbb{R}^3)$  be the subspace of  $H_h^1(\Omega, \mathbb{R}^3)$  with functions  $u \in H_h^1(\Omega, \mathbb{R}^3)$  vanishing on all boundary nodes  $\overline{\partial\Omega_D} \cap \{X_1, \dots, X_n\}$ . If vanishing of  $u$  on the nodes in  $\partial\Omega_D$  implies vanishing of  $u$  on  $\partial\Omega_D$ , we have  $H_{D,h}^1(\Omega, \mathbb{R}^3) = H_h^1(\Omega, \mathbb{R}^3) \cap H_D^1(\Omega, \mathbb{R}^3)$ . A solution  $u \in H_{D,h}^1(\Omega, \mathbb{R}^3)$  to the discretized elasticity PDE then fulfills

$$B(u, v) = \int_{\Omega} f \cdot v \, dx + \int_{\partial\Omega_N} g \cdot v \, dA, \forall v \in H_{D,h}^1(\Omega, \mathbb{R}^3). \quad (27)$$

Furthermore, such a solution always exists by the coercivity of the bilinear form  $B(u, v)$ , which also holds on  $H_{D,h}^1(\Omega, \mathbb{R}^3)$ .

In the derivation of the adjoint equation in Section 4 we need explicit expressions for both sides of (27) in terms of 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 rewrite (27) in terms of global variables  $U$  via

$$B(X)U = F(X), \quad B(X)_{(j,r),(k,s)} = B(e_r \theta_j, e_s \theta_k) \text{ and } F_{(j,r)} = \int_{\Omega} f \cdot e_r \theta_j \, dx + \int_{\partial\Omega_N} g \cdot e_r \theta_j \, dA, \quad (28)$$

with  $e_r$ ,  $r = 1, 2, 3$  the standard Basis on  $\mathbb{R}^3$ . The linear equation (28) is understood in the sense that the  $(k, s)$  indices of the stiffness matrix  $B(X)$  are contracted with the related indices of  $U$ . The solution of this equation is denoted by  $U(X)$ .

We have emphasised the dependency of the stiffness matrix  $B(X)$  and the load vector  $F(X)$  from the global node coordinate  $N \times 3$ -tensor  $X$ . The explicit calculations, which are completely standard, are given in Appendix A.2 in order to prepare the ground for the calculation of shape sensitivities.

In some situations, also the surface and volume force densities  $f = f(X)$  and  $g = g(X)$  are allowed to have a differentiable dependency on  $X$ . For  $f$  this makes sense e.g. in the case of centrifugal loads and we might want to adapt a surface force density  $g$  under geometry changes, in order to keep the total force acting on a part of the surface fixed, even if the surface volume changes. For notational simplicity, we will only introduce this when needed for a physically correct description.

## A.2 Discretization of the PDE

The integrals on both sides of (27) are implemented using a numerical quadrature with quadrature points  $\hat{\xi}_l$  and weights  $\hat{\omega}_l$ ,  $l = 1, \dots, l_q$ , on the reference element  $\hat{K}$ . We start

with the discretization of the left hand side of (27):

$$\begin{aligned}
B(u, v) &= \lambda \sum_{K \in \mathcal{T}_h} \int_K \nabla \cdot u \nabla \cdot v dx + 2\mu \sum_{K \in \mathcal{T}_h} \int_K \varepsilon(u) : \varepsilon(v) dx \\
&= \lambda \sum_{K \in \mathcal{T}_h} \int_{\widehat{K}} \nabla \cdot u(T_K(\widehat{\xi})) \nabla \cdot v(T_K(\widehat{\xi})) \det(\widehat{\nabla} T_K(\widehat{\xi})) d\widehat{\xi} \\
&\quad + 2\mu \sum_{K \in \mathcal{T}_h} \int_{\widehat{K}} \varepsilon(u(T_K(\widehat{\xi}))) : \varepsilon(v(T_K(\widehat{\xi}))) \det(\widehat{\nabla} T_K(\widehat{\xi})) d\widehat{\xi} \\
&= \lambda \sum_{K \in \mathcal{T}_h} \sum_{l=1}^{l_q} \widehat{\omega}_l \det(\widehat{\nabla} T_K(\widehat{\xi}_l)) \nabla \cdot u(T_K(\widehat{\xi}_l)) \nabla \cdot v(T_K(\widehat{\xi}_l)) \\
&\quad + 2\mu \sum_{K \in \mathcal{T}_h} \sum_{l=1}^{l_q} \widehat{\omega}_l \det(\widehat{\nabla} T_K(\widehat{\xi}_l)) \varepsilon(u(T_K(\widehat{\xi}_l))) : \varepsilon(v(T_K(\widehat{\xi}_l)))
\end{aligned} \tag{29}$$

After setting  $\omega_{lK} = \widehat{\omega}_l \det(\widehat{\nabla} T_K(\widehat{\xi}_l))$  and  $\xi_l^K = T_K(\widehat{\xi}_l)$ , this can be written as

$$B(u, v) = \underbrace{\lambda \sum_{K \in \mathcal{T}_h} \sum_{l=1}^{l_q} \omega_{lK} \nabla \cdot u(\xi_l^K) \nabla \cdot v(\xi_l^K)}_{B_1(u, v)} + \underbrace{2\mu \sum_{K \in \mathcal{T}_h} \sum_{l=1}^{l_q} \omega_{lK} \varepsilon(u(\xi_l^K)) : \varepsilon(v(\xi_l^K))}_{B_2(u, v)} \tag{30}$$

If we apply (15) and (16) to  $u$  and  $v$  and insert the result into (30), we obtain the discretization of the left hand side.

For the volume integral on the right hand side of (27) we obtain by a similar argument

$$\int_{\Omega} f \cdot v dx = \sum_{K \in \mathcal{T}_h} \sum_{l=1}^{l_q} \omega_{lK} f(\xi_l^K) \cdot v(\xi_l^K). \tag{31}$$

And finally we get the following expression for the discretized surface integral

$$\int_{\partial\Omega} g \cdot v dA = \sum_{F \in \mathcal{N}_h} \sum_{l=1}^{l_q^F} \omega_{lF} g(\xi_{lF}) \cdot v(\xi_{lF}), \text{ where } \omega_{lF} = \widehat{\omega}_l^F \sqrt{\det g_F(\widehat{\xi}_l^F)} \text{ and } \xi_l^F = T_{K(F)}(\widehat{\xi}_l^F). \tag{32}$$

Here  $\widehat{\xi}_l^F$  and  $\widehat{\omega}_l^F$  are quadrature points on a reference face  $\widehat{F}$  of the reference element  $\widehat{K}$  and  $\mathcal{N}_h$  is the collection of all finite element faces that lie in  $\partial\Omega$ .  $g_F(\widehat{\xi}) = \widehat{\nabla}^{\widehat{F}}(T_K \upharpoonright \widehat{F})(\widehat{\xi}) \widehat{\nabla}^{\widehat{F}}(T_K \upharpoonright \widehat{F})^T(\widehat{\xi})$  is the Gram matrix on  $\widehat{F}$ , where  $T_K : \widehat{F} \rightarrow F$ .

If the selected quadratures are not exact, the above identities have to be understood in the sense of approximations.

### A.3 Computing $\frac{\partial J}{\partial U}$

In the following, we use some conventions related to the connectivity mapping  $\widehat{j} : \mathcal{T}_h \times \{1, \dots, n_{\text{sh}}\} \rightarrow \{1, \dots, N\}$ . For  $K \in \mathcal{T}_h$ , we denote by  $\widehat{j}_K : \{1, \dots, n_{\text{sh}}\} \rightarrow \{1, \dots, N\}$

the restriction of the connectivity mapping  $\widehat{j}$  to the set  $\{(K, 1), \dots, (K, n_k)\}$ , where we identify  $1, \dots, n_{\text{sh}}$  with  $(K, 1), \dots, (K, n_{\text{sh}})$ .

For a fixed global index  $j \in \{1, 2, \dots, N\}$  we have  $\widehat{j}^{-1}(j) = \{(K_1, m_1), \dots, (K_f, m_f)\}$ , where  $K_1, \dots, K_f \in \mathcal{T}_h$  and  $m_1, \dots, m_f \in \{1, 2, \dots, n_{\text{sh}}\}$ . With  $\widehat{j}^{-1}(j)_1 = \{K_1, \dots, K_f\}$  we denote the set projection to the first component.

For  $k = 1, 2, 3$  let  $u_{jk}$  be the  $x, y, z$  coordinate of the global degree of freedom  $u_j$ . We obtain for the partial derivative of  $J$  with respect to this variable

$$\begin{aligned} \frac{\partial J(X, U)}{\partial u_{jk}} &= \sum_{F \in \mathcal{N}_h} \sum_{l=1}^{l_q^F} \omega_{lF} \\ &\times \frac{\partial}{\partial u_{jk}} \left[ \text{CMB}^{-1} \left( \varepsilon_a^{\text{el-pl}} \left( \sum_{m=1}^{n_{\text{sh}}} u_{\widehat{j}(K(F), m)} \otimes ((\widehat{\nabla} T_{K(F)}^T(\widehat{\xi}_l^F))^{-1} \widehat{\nabla} \widehat{\theta}_m(\widehat{\xi}_l^F)) \right) \right) \right]^{-\bar{m}}. \end{aligned} \quad (33)$$

We will frequently use the abbreviation

$$q = q_{lF} = q_{lF}(U, X) = \nabla u(\widehat{\xi}_l^F) = \sum_{m=1}^{n_{\text{sh}}} u_{\widehat{j}(K(F), m)} \otimes (\widehat{\nabla} T_{K(F)}^T(\widehat{\xi}_l^F))^{-1} \widehat{\nabla} \widehat{\theta}_m(\widehat{\xi}_l^F), \quad l = 1, \dots, l_q^F, \quad (34)$$

with  $\widehat{\xi}_l^F = T_{K(F)}(\widehat{\xi}_l)$ . We thus have

$$\frac{\partial}{\partial u_{jk}} [\text{CMB}^{-1}(\varepsilon_a^{\text{el-pl}}(q_{lF}))]^{-\bar{m}} = -\bar{m} [\text{CMB}^{-1}(\varepsilon_a^{\text{el-pl}}(q_{lF}))]^{-\bar{m}-1} \frac{\partial}{\partial u_{jk}} [\text{CMB}^{-1}(\varepsilon_a^{\text{el-pl}}(q_{lF}))]. \quad (35)$$

In the next step, we compute

$$\frac{\partial}{\partial u_{jk}} [\text{CMB}^{-1}(\varepsilon_a^{\text{el-pl}}(q_{lF}))] = \frac{\frac{\partial}{\partial u_{jk}} (\varepsilon_a^{\text{el-pl}}(q_{lF}))}{\frac{\partial \text{CMB}}{\partial \varepsilon_a^{\text{el-pl}}} (\text{CMB}^{-1}(\varepsilon_a^{\text{el-pl}}(q_{lF})))} \quad (36)$$

We now calculate

$$\frac{\partial}{\partial u_{jk}} (\varepsilon_a^{\text{el-pl}}(q_{lF})) = \frac{\partial \varepsilon_a^{\text{el-pl}}(q_{lF})}{\partial q_{lF}} : \frac{\partial q_{lF}}{\partial u_{jk}} = \text{tr} \left( \left( \frac{\partial \varepsilon_a^{\text{el-pl}}(q_{lF})}{\partial q_{lF}} \right)^T \frac{\partial q_{lF}}{\partial u_{jk}} \right), \quad (37)$$

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

$$\begin{aligned} \frac{\partial (q_{lF})_{sn}}{\partial u_{jk}} &= \frac{\partial}{\partial u_{jk}} \left( \sum_{m=1}^{n_{\text{sh}}} u_{\widehat{j}(K(F), m)_s} \left( (\widehat{\nabla} T_{K(F)}^T(\widehat{\xi}_l^F))^{-1} \widehat{\nabla} \widehat{\theta}_m(\widehat{\xi}_l^F) \right)_n \right) \\ &= \sum_{m=1}^{n_{\text{sh}}} \frac{\partial u_{\widehat{j}(K(F), m)_s}}{\partial u_{jk}} \left( (\widehat{\nabla} T_{K(F)}^T(\widehat{\xi}_l^F))^{-1} \widehat{\nabla} \widehat{\theta}_m(\widehat{\xi}_l^F) \right)_n \\ &= \begin{cases} \delta_{sk} \left( (\widehat{\nabla} T_{K(F)}^T(\widehat{\xi}_l^F))^{-1} \widehat{\nabla} \widehat{\theta}_{\widehat{j}^{-1}(j)}(\widehat{\xi}_l^F) \right)_n & \text{if } j \in \widehat{j}_{K(F)}(\{1, \dots, n_{\text{sh}}\}) \\ 0 & \text{otherwise} \end{cases}, \end{aligned} \quad (38)$$

with  $\delta_{sk}$  Kronecker's symbol. In order to simplify the calculation, we square (5) and divide by  $E$ . The resulting relation between  $\varepsilon_a^{\text{el-pl}}$  and  $\frac{\sigma_a^2}{E}$  then is denoted by  $\varepsilon_a^{\text{el-pl}} = \bar{\text{SD}}\left(\frac{\sigma_a^2}{E}\right)$ . The partial derivative of  $\varepsilon_a^{\text{el-pl}}$  with respect to  $q_{sm}$  is now:

$$\frac{\partial \varepsilon_a^{\text{el-pl}}}{\partial q_{sm}} = \frac{d\text{RO} \circ \bar{\text{SD}}^{-1}\left(\frac{(\sigma_a)^2}{E}\right)}{d\left(\frac{(\sigma_a)^2}{E}\right)} \cdot \frac{\partial}{\partial q_{sm}}\left(\frac{(\sigma_a)^2}{E}\right). \quad (39)$$

The derivative of the function  $\text{RO} \circ \bar{\text{SD}}^{-1}(\cdot)$  is easily calculated from (3) and (5). We further calculate

$$\frac{\partial}{\partial q_{sm}}\left(\frac{(\sigma_a)^2}{E}\right) = \frac{3\mu^2}{4E} \sum_{i,j=1}^3 A_{ij} \frac{\partial A_{ij}}{\partial q_{sm}} \quad (40)$$

using  $A_{ij} = \left(-\frac{2}{3}\left(\frac{1}{2}(q_{ij} + q_{ji})\right)\delta_{ij} + q_{ij} + q_{ji}\right)$ . We thus obtain

$$\frac{\partial A_{ij}}{\partial q_{s,m}} = -\frac{2}{3}\delta_{ij}\delta_{sm} + \delta_{is}\delta_{jm} + \delta_{js}\delta_{im}. \quad (41)$$

We thus get for the left hand side of (40)

$$\frac{3\mu^2}{4E} \sum_{i,j=1}^3 \left(-\frac{2}{3}\left(\frac{1}{2}(q_{ij} + q_{ji})\right)\delta_{ij} + q_{ij} + q_{ji}\right) \left(-\frac{2}{3}\delta_{ij}\delta_{sm} + \delta_{is}\delta_{jm} + \delta_{js}\delta_{im}\right). \quad (42)$$

#### A.4 Computing $\frac{\partial J}{\partial X}$

As in Subsection A.3, We use the abbreviation  $q_{lF}$  for  $\nabla u(\xi_l^F)$ , see (34). The partial derivative of  $J(X, U)$  w.r.t. the global  $j$ th mesh node  $i$ -coordinate  $X_{ji}$ ,  $i = 1, 2, 3$  and  $j = 1, \dots, N$ , is

$$\begin{aligned} \frac{\partial J(X, U)}{\partial X_{ji}} &= \sum_{F \in \mathcal{N}_h} \sum_{l=1}^{l_q^F} \frac{\partial}{\partial X_{ji}} \left( \omega_{lF} \left( \text{CMB}^{-1} \left( \varepsilon_a^{\text{el-pl}}(q_{lF}) \right) \right)^{-\bar{m}} \right) \\ &= \sum_{F \in \mathcal{N}_h} \sum_{l=1}^{l_q^F} \left[ \frac{\partial \omega_{lF}}{\partial X_{ji}} \left( \text{CMB}^{-1} \left( \varepsilon_a^{\text{el-pl}}(q_{lF}) \right) \right)^{-\bar{m}} \right. \\ &\quad \left. + \omega_{lF} \frac{\partial}{\partial X_{ji}} \left( \left( \text{CMB}^{-1} \left( \varepsilon_a^{\text{el-pl}}(q_{lF}) \right) \right)^{-\bar{m}} \right) \right]. \end{aligned} \quad (43)$$

We compute first the partial derivative of  $\frac{\partial \omega_{lF}}{\partial X_{ji}}$  as

$$\begin{aligned} \frac{\partial \omega_{lF}}{\partial X_{ji}} &= \frac{\partial}{\partial X_{ji}} \left( \hat{\omega}_l \sqrt{\det(g_F(\hat{\xi}_l^F))} \right) \\ &= \hat{\omega}_l \frac{1}{2} (\det(g_F(\hat{\xi}_l^F)))^{-1/2} \frac{\partial}{\partial X_{ji}} (\det(g_F(\hat{\xi}_l^F))). \end{aligned} \quad (44)$$

The derivative of the determinant is

$$\frac{\partial}{\partial X_{ji}} \left( \det \left( g_F(\widehat{\xi}_l^F) \right) \right) = \det \left( g_F(\widehat{\xi}_l^F) \right) \operatorname{tr} \left( \left( g_F(\widehat{\xi}_l^F) \right)^{-1} \frac{\partial g_F(\widehat{\xi}_l^F)}{\partial X_{ji}} \right), \quad (45)$$

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

$$J_F(\widehat{\xi}_l^F) = \frac{\partial T_K(\widehat{\xi}_l^F)}{\partial \widehat{X}^F} = \sum_{r=1}^{n_{\text{sh}}} \begin{pmatrix} X_{r1}^{K(F)} \frac{\partial \widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X}_1^F} & X_{1r}^{K(F)} \frac{\partial \widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X}_2^F} \\ X_{r2}^{K(F)} \frac{\partial \widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X}_1^F} & X_{r2}^{K(F)} \frac{\partial \widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X}_2^F} \\ X_{r3}^{K(F)} \frac{\partial \widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X}_1^F} & X_{r3}^{K(F)} \frac{\partial \widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X}_2^F} \end{pmatrix}. \quad (46)$$

Here  $\widehat{X}_i^F$ ,  $i = 1, 2$ , are the coordinates on the two dimensional reference face  $\widehat{F}$  corresponding to  $F$  in  $\widehat{K}$ .

The derivative  $\frac{\partial g_F(\widehat{\xi}_l^F)}{\partial X_{ji}}$  is thus given by

$$\frac{\partial g_F(\widehat{\xi}_l^F)}{\partial X_{ji}} = \frac{\partial}{\partial X_{ji}} (J_F(\widehat{\xi}_l^F))^T J_F(\widehat{\xi}_l^F) + (J_F(\widehat{\xi}_l^F))^T \frac{\partial}{\partial X_{ji}} (J_F(\widehat{\xi}_l^F)). \quad (47)$$

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

$$\begin{aligned} \frac{\partial}{\partial X_{ji}} (J_F(\widehat{\xi}_l^F)_{sk}) &= \sum_{r=1}^{n_{\text{sh}}} \frac{\partial}{\partial X_{ji}} \left( X_{rs}^{K(F)} \frac{\partial \widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X}_k} \right) \\ &= \begin{cases} \delta_{is} \frac{\partial \widehat{\theta}_{\widehat{j}_{K(F)}(j)}(\widehat{\xi}_l^F)}{\partial \widehat{X}_k} & \text{if } j \in \widehat{j}_{K(F)}(\{1, \dots, n_{\text{sh}}\}) \\ 0 & \text{otherwise} \end{cases}. \end{aligned} \quad (48)$$

This finishes the computation of the first term on the right hand side of (43).

To compute the second term in (43), we take the partial derivative

$$\frac{\partial}{\partial X_{ji}} \left[ \text{CMB}^{-1}(\varepsilon_a^{\text{el-pl}}(q_{lF})) \right]^{-\bar{m}} = \frac{-\bar{m} [\text{CMB}^{-1}(\varepsilon_a^{\text{el-pl}}(q_{lF}))]^{-\bar{m}-1}}{\frac{\partial \text{CMB}}{\partial \varepsilon_a^{\text{el-pl}}}(\text{CMB}^{-1}(\varepsilon_a^{\text{el-pl}}(q_{lF})))} \frac{\partial}{\partial X_{ji}} (\varepsilon_a^{\text{el-pl}}(q_{lF})), \quad (49)$$

with

$$\frac{\partial}{\partial X_{ji}} (\varepsilon_a^{\text{el-pl}}(q_{lF})) = \frac{\partial \varepsilon_a^{\text{el-pl}}(q_{lF})}{\partial q_{lF}} : \frac{\partial q_{lF}}{\partial X_{ji}}. \quad (50)$$

In the above equation,  $:$  stands for the contraction of both  $q$  indices. Next we have to compute

$$\begin{aligned} \frac{\partial q_{lF}}{\partial X_{ji}}(X) &= \frac{\partial}{\partial X_{ji}} \left( \sum_{m=1}^{n_{\text{sh}}} u_{\widehat{j}(K(F),m)} \otimes ((\widehat{\nabla} T_{K(F)}^T(\widehat{\xi}_l^F))^{-1} \widehat{\nabla} \widehat{\theta}_m(\widehat{\xi}_l^F)) \right) \\ &= \sum_{m=1}^{n_{\text{sh}}} u_{\widehat{j}(K(F),m)} \otimes \left( \frac{\partial}{\partial X_{ji}} (\widehat{\nabla} T_{K(F)}^T(\widehat{\xi}_l^F))^{-1} \right) \widehat{\nabla} \widehat{\theta}_m(\widehat{\xi}_l^F), \end{aligned} \quad (51)$$

where

$$\frac{\partial}{\partial X_{ji}} (\widehat{\nabla} T_K^T(\widehat{\xi}_l^F))^{-1} = -(\widehat{\nabla} T_K^T(\widehat{\xi}_l^F))^{-1} \frac{\partial (\widehat{\nabla} T_K^T(\widehat{\xi}_l^F))}{\partial X_{ji}} (\widehat{\nabla} T_K^T(\widehat{\xi}_l^F))^{-1}. \quad (52)$$

The Jacobian matrix has the form

$$\widehat{\nabla} T_K(\widehat{\xi}_l^F) = \begin{pmatrix} \frac{\partial X_1^K}{\partial \widehat{X}_1} & \frac{\partial X_1^K}{\partial \widehat{X}_2} & \frac{\partial X_1^K}{\partial \widehat{X}_3} \\ \frac{\partial X_2^K}{\partial \widehat{X}_1} & \frac{\partial X_2^K}{\partial \widehat{X}_2} & \frac{\partial X_2^K}{\partial \widehat{X}_3} \\ \frac{\partial X_3^K}{\partial \widehat{X}_1} & \frac{\partial X_3^K}{\partial \widehat{X}_2} & \frac{\partial X_3^K}{\partial \widehat{X}_3} \end{pmatrix} = \sum_{r=1}^{n_{\text{sh}}} \begin{pmatrix} X_{r1}^K \frac{\partial \widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X}_1} & X_{r1}^K \frac{\partial \widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X}_2} & X_{r1}^K \frac{\partial \widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X}_3} \\ X_{r2}^K \frac{\partial \widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X}_1} & X_{r2}^K \frac{\partial \widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X}_2} & X_{r2}^K \frac{\partial \widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X}_3} \\ X_{r3}^K \frac{\partial \widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X}_1} & X_{r3}^K \frac{\partial \widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X}_2} & X_{r3}^K \frac{\partial \widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X}_3} \end{pmatrix}. \quad (53)$$

Finally, we get

$$\frac{\partial (\widehat{\nabla} T_K^T(\widehat{\xi}_l^F)_{sk})}{\partial X_{ji}} = \sum_{r=1}^{n_{\text{sh}}} \frac{\partial X_{rk}^K}{\partial X_{ji}} \frac{\partial \widehat{\theta}_r(\widehat{\xi}_l^F)}{\partial \widehat{X}_s} = \begin{cases} \delta_{ik} \frac{\partial \widehat{\theta}_{\widehat{j}_K^{-1}(j)}(\widehat{\xi}_l^F)}{\partial \widehat{X}_s}, & \text{if } j \in \widehat{j}_K(\{1, \dots, n_{\text{sh}}\}) \\ 0 & \text{otherwise} \end{cases}. \quad (54)$$

This finishes the computation of the second term.

## A.5 Computing $\frac{\partial B}{\partial X}$

As in (29) we have  $B(X)_{(q,r),(k,s)} = B(e_r \theta_q, e_s \theta_k) = B_1(e_r \theta_q, e_s \theta_k) + B_2(e_r \theta_q, e_s \theta_k)$ , then  $\frac{\partial B(X)_{(q,r),(k,s)}}{\partial X} = \frac{\partial B_1(e_r \theta_q, e_s \theta_k)}{\partial X_{ji}} + \frac{\partial B_2(e_r \theta_q, e_s \theta_k)}{\partial X_{ji}}$ . For the first partial derivative, one obtains

$$\begin{aligned} \frac{\partial B_1(e_r \theta_q, e_s \theta_k)}{\partial X_{ji}} = \lambda \sum_{K \in \widehat{j}^{-1}(j)_1 \cap \widehat{j}^{-1}(q)_1 \cap \widehat{j}^{-1}(k)_1} \sum_{l=1}^{l_q} \left\{ \frac{\partial}{\partial X_{ji}} (\omega_{lK}) \nabla_r \theta_q(\xi_l^K) \nabla_s \theta_k(\xi_l^K) \right. \\ \left. + \omega_{lK} \frac{\partial}{\partial X_{ji}} (\nabla_r \theta_q(\xi_l^K)) \nabla_s \theta_k(\xi_l^K) \right. \\ \left. + \omega_{lK} \nabla_r \theta_q(\xi_l^K) \frac{\partial}{\partial X_{ji}} (\nabla_s \theta_k(\xi_l^K)) \right\}. \end{aligned} \quad (55)$$

Here we use the convention that  $\widehat{j}^{-1}(j)_1$  is the projection of the set  $\widehat{j}^{-1}(j)$  to the first component (the index of the element).

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

$$\frac{\partial}{\partial X_{ji}} (\omega_{lK}) = \frac{\partial}{\partial X_{ji}} (\widehat{\omega}_l \det(\widehat{\nabla} T_K(\widehat{\xi}_l))) = \widehat{\omega}_l \frac{\partial}{\partial X_{ji}} (\det(\widehat{\nabla} T_K(\widehat{\xi}_l))) \quad (56)$$

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

$$\frac{\partial}{\partial X_{ji}} (\omega_{lK}) = \widehat{\omega}_l \det(\widehat{\nabla} T_K(\widehat{\xi}_l)) \text{tr} \left( (\widehat{\nabla} T_K(\widehat{\xi}_l))^{-1} \frac{\partial \widehat{\nabla} T_K(\widehat{\xi}_l)}{\partial X_{ji}} \right). \quad (57)$$

The partial  $X_{ji}$  derivative of the matrix  $\widehat{\nabla}T_K(\widehat{\xi}_l)$  has been calculated in (54), where we have to replace  $\widehat{\xi}_l^F$  with  $\widehat{\xi}_l$ .

The second partial derivative in  $\frac{\partial B_1}{\partial X_{ji}}$  is easily calculated

$$\frac{\partial \nabla_r \theta_q}{\partial X_{ji}}(\xi_l^K) = \frac{\partial}{\partial X_{ji}} \left[ \left( \widehat{\nabla}T_K(\widehat{\xi}_l)^T \right)^{-1} \widehat{\nabla} \widehat{\theta}_{\widehat{j}_K^{-1}(q)}(\widehat{\xi}_l) \right]_r. \quad (58)$$

We now refer to equations (52) to (54) in Section A.4 to conclude the computation. Here, of course, the surface quadrature point  $\widehat{\xi}_l^F$  has to be replaced by the volume quadrature point  $\widehat{\xi}_l$ . The third partial derivative in (55) is completely analogous to the second.

For the partial derivative  $\frac{\partial B_2}{\partial X}$  we obtain

$$\begin{aligned} \frac{\partial B_2(e_r \theta_q, e_s \theta_k)}{\partial X_{ji}} &= 2\mu \sum_{K \in \widehat{j}^{-1}(j)_1 \cap \widehat{j}^{-1}(q)_1 \cap \widehat{j}^{-1}(k)_1} \sum_{l=1}^{l_q} \\ &\left\{ \frac{\partial}{\partial X_{ji}} (\omega_{lK}) \varepsilon(e_r \theta_q(\xi_l^K)) : \varepsilon(e_s \theta_k(\xi_l^K)) \right. \\ &+ \omega_{lK} \frac{\partial}{\partial X_{ji}} (\varepsilon(e_r \theta_q(\xi_l^K))) : \varepsilon(e_s \theta_k(\xi_l^K)) \\ &\left. + \omega_{lK} \varepsilon(e_r \theta_q(\xi_l^K)) : \frac{\partial}{\partial X_{ji}} (\varepsilon(e_s \theta_k(\xi_l^K))) \right\}. \end{aligned} \quad (59)$$

The first term is calculated in (56). For the second term, we observe that the linear elastic strain tensor field is given by

$$\varepsilon(e_r \theta_q(\xi_l^K))_{ab} = \frac{1}{2} (\delta_{rb} \nabla_a \theta_q(\xi_l^K) + \delta_{ra} \nabla_b \theta_q(\xi_l^K)), \quad (60)$$

and we refer to the argument following Eq. (58) to conclude the computation of (59).

## A.6 Computing $\frac{\partial F}{\partial X}$

We have  $\frac{\partial F_{(q,r)}}{\partial X_{ji}} = \frac{\partial F_{(q,r)}^{\text{vol}}}{\partial X_{ji}} + \frac{\partial F_{(q,r)}^{\text{surf}}}{\partial X_{ji}}$ . Starting with the volume term, we obtain

$$\begin{aligned} \frac{\partial F_{(q,r)}^{\text{vol}}}{\partial X_{ji}} &= \sum_{K \in \widehat{j}^{-1}(j)_1 \cap \widehat{j}^{-1}(q)_1} \sum_{l=1}^{l_q} \left\{ \frac{\partial}{\partial X_{ji}} (\omega_{lK}) f_r(\xi_l^K) \theta_q(\xi_l^K) \right. \\ &\left. + \omega_{lK} \frac{\partial}{\partial X_{ji}} (f_r(\xi_l^K)) \theta_q(\xi_l^K) + \omega_{lK} f_r(\xi_l^K) \frac{\partial}{\partial X_{ji}} (\theta_q(\xi_l^K)) \right\} \end{aligned} \quad (61)$$

The partial derivative of the volume weight has been calculated in (56). The third term in (61) vanishes, as  $\theta_q(\theta_l) = \widehat{\theta}_{\widehat{j}_K^{-1}(q)}(\widehat{\xi}_l)$  does not depend on  $X_{ji}$ . Unless the volume force density  $f$  does depend explicitly on the position (like in the case of centrifugal loads) this term vanishes as the third one.

Finally we have to calculate the partial derivative of the surface loads

$$\begin{aligned} \frac{\partial F_{(q,r)}^{\text{surf}}}{\partial X_{ji}} = & \sum_{\substack{F \in \mathcal{N}_h \\ K(F) \in \hat{\mathcal{J}}^{-1}(j)_1 \cap \hat{\mathcal{J}}^{-1}(q)_1}} \sum_{l=1}^{l_q^F} \left\{ \frac{\partial}{\partial X_{ji}} (\omega_{lF}) f_r(\xi_l^F) \theta_q(\xi_l^F) \right. \\ & \left. + \omega_{lF} \frac{\partial}{\partial X_{ji}} (f_r(\xi_l^F)) \theta_q(\xi_l^F) + \omega_{lF} f_r(\xi_l^F) \frac{\partial}{\partial X_{ji}} (\theta_q(\xi_l^F)) \right\}. \end{aligned} \quad (62)$$

The first term can be calculated with the aid of (44). For the second and third term, the same reasoning applies as for partial derivative of the volume force.

## References

- [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. Bonnetier, G. Francfort and F. Jouve, *Shape Optimization by the Homogenization Method*, Numer. Math., **76**, 1997, 27-68.
- [4] I. Babuška, Z. Sawlan, M. Scavino B. Szabó and R. Tempone, Spatial Poisson processes for crack initiation, (2018) arXiv:1805.03433.
- [5] W. Bangerth, R. Rannacher, *Adaptive Finite Element Methods for Differential Equations*, Birkhäuser 2003.
- [6] 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.
- [7] 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.
- [8] 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)
- [9] M. Bolten, C. Hahn, H. Gottschalk and M. Saadi, Numerical shape optimization to decrease the failure probability of ceramic structures, (2017) arXiv:1705.05776.
- [10] M. Bolten, H. Gottschalk and S. Schmitz, *Minimal failure probability for ceramic design via shape control*, J. Optim. Theory Appl., **166** (2015), 983-1001.
- [11] 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.

- [12] A. Borzi and V. Schulz, *Computational Optimization of Systems governed by Partial Differential Equations*, SIAM series on computational engineering, SIAM 2012.
- [13] D. Braess, *Finite Elemente - Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie*, fourth edition, Springer, Berlin, 2007.
- [14] P. Ciarlet, *Mathematical Elasticity - Volume I: Three-Dimensional Elasticity*, Studies in Mathematics and its Applications, Vol. 20, North-Holland, Amsterdam, 1988.
- [15] A. Ern and J.-L. Guermond, *Theory and Practice of Finite Elements*, Springer, New York, 2004.
- [16] L. A. Escobar and W. Q. Meeker, *Statistical Methods for Reliability Data*, Wiley-Interscience Publication, New York, 1998.
- [17] D. Munz and T. Fett, *Ceramics*, (engl. edition) Springer Verlag Berlin, 2012.
- [18] H. Gottschalk, M. Saadi, O. T. Doganay, K. Klamroth and S. Schmitz, Adjoint Method to Calculate Shape Gradients of Failure Probabilities for Turbomachinery Components, ASME-Turbo-Expo GT2018-75759 (2018).
- [19] 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.
- [20] H. Gottschalk, S. Schmitz, T. Seibel, R. Krause, G. Rollmann and T. Beck, *Probabilistic Schmid Factors and Scatter of LCF Life*, Materials Science and Engineering 46 (2) (2015), 156-164..
- [21] G. Gottstein, *Physical foundations of Material Science*, Springer 2004.
- [22] J. Haslinger and R. A. E. Mäkinen, *Introduction to Shape Optimization - Theory, Approximation and Computation*, SIAM - Advances in Design and Control, 2003.
- [23] Hertel, O., Vormwald, M., *Statistical and geometrical size effects in notched members based on weakest-link and short-crack modelling*, Engineering Fracture Mechanics, **95** (2012), pp 72–83.
- [24] 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.
- [25] O. Kallenberg, *Random Measures*, Akademie-Verlag, Berlin 1983.
- [26] 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.
- [27] W. B. Liu, P. Neittaanmäki and D. Tiba, *Existence for shape optimization problems in arbitrary dimension*, SIAM J. Contr. Optimization **41** (2003) 1440-1454.
- [28] 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.

- [29] L. Mäde, S. Schmitz, H. Gottschalk and T. Beck, Combined notch and size effect modeling in a local probabilistic approach for LCF, *Comp. Materials Science* 142 (2018) 377388.
- [30] H. Neuber, *Theory of Stress Concentration for Shear-Strained Prismatical Bodies with Arbitrary Nonlinear Stress-Strain Law*, *J. Appl. Mech.* 26, 544, 1961.
- [31] D. Radaj and M. Vormwald, *Ermüdungsfestigkeit*, third edition, Springer, Berlin Heidelberg, 2007.
- [32] 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.
- [33] H. Riesch-Oppermann, A. Brückner-Foit, 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)
- [34] S. Schmitz, *A Probabilistic Local Model for Low Cycle Fatigue – New Aspects of Structural Mechanics*, Dissertation Lugano and Wuppertal 2014, appeared in Hartung-Gorre Verlag, 2014.
- [35] 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.
- [36] 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).
- [37] 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.
- [38] J. Sokolowski and J.-P. Zolesio, *Introduction to Shape Optimization - Shape Sensitivity Analysis*, first edition, Springer, Berlin Heidelberg, 1992.
- [39] 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.
- [40] F. Tröltzsch, *Optimal Control of Partial Differential Equations (in German)*, Vieweg + Teubner, 2010.
- [41] S. Watanabe, *On discontinuous additive functionals and Lvy measures of a Markov process*, *Japan. J. Math.* 34 (1964).
- [42] E. W. Weibull, *A Statistical Theory of the Strength of Materials*, *Ingeniors Vetenskaps Akad. Handl.* 151, 1–45 (1939)
- [43] K. Wittig, *Construction of a gas turbine for model air planes (in German)*, Munich 1993, www.calculix.de.