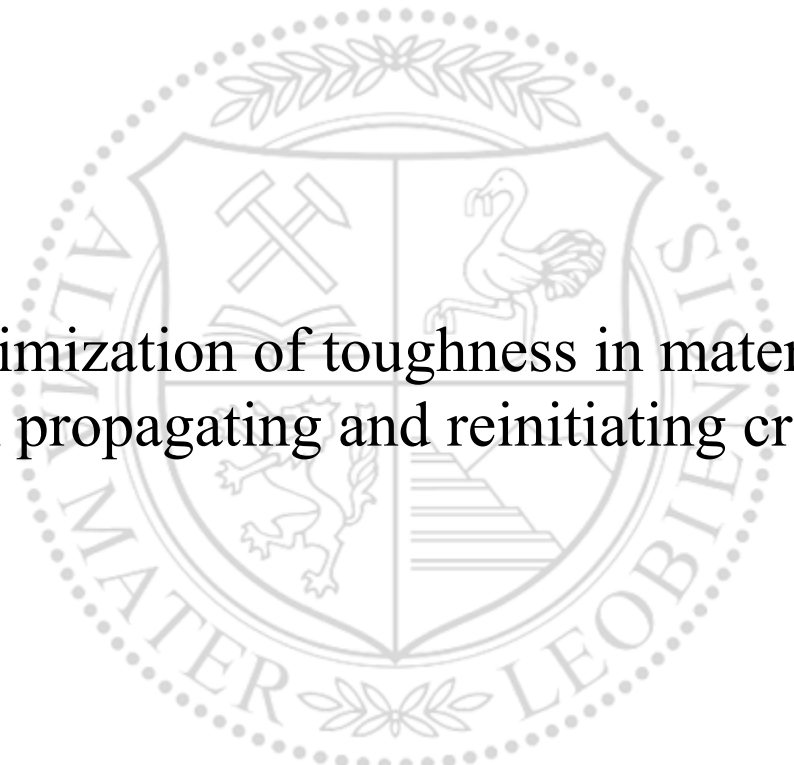




Chair of Designing Plastics and Composite Materials

Doctoral Thesis



Optimization of toughness in materials  
with propagating and reinitiating cracks

Dipl.-Ing. Matthias Retzl, BSc

October 2024





## **AFFIDAVIT**

I declare on oath that I wrote this thesis independently, did not use any sources and aids other than those specified, have fully and truthfully reported the use of generative methods and models of artificial intelligence, and did not otherwise use any other unauthorized aids.

I declare that I have read, understood and complied with the "Preamble on Integrity in Academic Study, Teaching, and Research Operations" of the Montanuniversität Leoben.

Furthermore, I declare that the electronic and printed versions of the submitted thesis are identical in form and content.

Date

(The original signature is kept at the university)



# Declaration of the use of artificial intelligence

I used the following tools based on artificial intelligence:

subject	AI share (in %)	AI tool version	remark	reference to prompting
Creating tables	1	Copilot 2024	Generation of a latex table for the declaration of the use of AI based tools	<a href="https://oc.unileoben.ac.at/index.php/s/ZQ0BDTCD4CUC22U">oc.unileoben.ac.at/index.php/s/ZQ0BDTCD4CUC22U</a>
Improving linguistic readability	30	deepl 2024	30% of the thesis was checked using deepl	n/a
Research of relevant literature citations	5	Copilot 2024	Help with the literature search, but independent development of the central statements	<a href="https://oc.unileoben.ac.at/index.php/s/cJLhDIyY0ud2UKz">oc.unileoben.ac.at/index.php/s/cJLhDIyY0ud2UKz</a>



# Acknowledgements

This work was conducted out at the Chair of Designing Plastics and Composite Materials at Montanuniversität Leoben. First, I want to thank Univ.-Prof. Clara Schuecker, for the opportunity to work at the chair and to write my PhD thesis. I am grateful for the support and the guidance throughout the whole project as my academic supervisor.

Special thanks go to Dr. Martin Pletz for all the vivid discussions that helped to stay focused. Martin was my main supervisor who helped me out with valuable input and inspiring discussions. He also reviewed all my work very fast in many iterations such that I was able to quickly adapt it. Thanks for sharing the experience and skills to perform this research.

I want to thank Univ.-Prof. Clemens Brand for being my mentor. Through his lectures, I developed a deep understanding for the Finite Element Method.

I also thank Priv.-Doz. Benoît Loridant for the discussions about tensors. It helped me to understand how configurational forces work.

Further thanks go to my colleague Christoph Waly from our neighbor chair of Materials Science and Testing of Polymers. We discussed how to experimentally test the investigated theoretically methods.

I would also like to thank my colleagues from my chair who were always willing to help when it came to issues with drawing graphics and writing papers. Furthermore, I enjoyed spending my spare time with them when we went climbing with Matthias Drovderic and Martin Hubmann. My colleague Vasco always empowered my self-confidence by letting me win most of our chess games and my colleague Christian Moser was always available for a chat. Our secretary Heike was very helpful with all the organizational stuff and for mental support.

It was an honor to supervise my student workers Maximilian Pettinger, Julia Schneider and Mariia Portnova and guide them through their first steps in Python with object-oriented programming.

Of course, I do not want to miss the opportunity to thank my family and friends, especially my parents Herbert and Regina and my sister Sophia who always supported me during my studies and provided me company during the Corona crises.





# Abstract

In many technical applications, materials need to absorb energy or be tolerant towards flaws that may already be generated in the manufacturing process. This behavior is known as toughness, and the fracture toughness is a widely known measure that allows to compare various materials to each other. Solid materials can be tough if they form a plastic zone, but in nature one can also observe hierarchical structures of very brittle base materials, where the overall mechanical response is much tougher than the brittle base material. This work aims to create such structures with a high toughness. To obtain such a structure, homogeneous 2D plates with arbitrarily shaped holes are considered. The goal is to optimize the shape of the holes, but first the fracture process must be predicted. During the fracture process, new cracks can initiate from surfaces and existing cracks may propagate. Crack initiation from a surface can be predicted using Leguillon's Coupled Criterion or the Theory of Critical Distances of Taylor. For the crack propagation, two parameters need to be predicted: the critical load at which a crack will propagate and the direction of the crack propagation. According to Griffith, a load becomes critical when the energy release rate exceeds the critical energy release rate which is a material parameter. A common criterion for the direction of the crack propagation is the Maximum Energy Release Rate (MERR) criterion. In this work, Configurational Forces are implemented as an Abaqus plugin. This implementation can be used to predict both, the critical load and the direction of the crack initiation. This allows to simulate the fracture process and the tensile toughness of the overall structure can be computed. Furthermore, an optimization algorithm is developed to maximize the tensile toughness. For an example problem, a hole structure is found with a tensile toughness more than 4.5 times higher compared to a solid material.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Objective of the thesis</b>	<b>3</b>
<b>3</b>	<b>Prediction of fracture</b>	<b>7</b>
3.1	Failure criteria . . . . .	9
3.2	Linear Elastic Fracture Mechanics . . . . .	10
3.2.1	Griffith's G-concept . . . . .	10
3.2.2	Irwin's K-concept . . . . .	11
3.2.3	Mixed mode I-II . . . . .	12
3.2.4	Computation of the energy release rate . . . . .	14
3.3	Finite Fracture Mechanics . . . . .	21
3.3.1	Theory of Critical Distances . . . . .	22
3.3.2	Energy Criterion . . . . .	23
3.3.3	Coupled Criterion . . . . .	24
3.3.4	Computation of the incremental energy release rate . . . . .	27
<b>4</b>	<b>Toughness optimization</b>	<b>31</b>
4.1	Optimization algorithm . . . . .	35
4.2	Representation of hole shapes . . . . .	36
4.3	Optimization result . . . . .	37
<b>5</b>	<b>Compilation of the thesis</b>	<b>39</b>
<b>6</b>	<b>Conclusions</b>	<b>41</b>
<b>7</b>	<b>References</b>	<b>43</b>
	<b>Papers</b>	<b>55</b>
	Paper A . . . . .	55
	Paper B . . . . .	73
	Paper C . . . . .	81
	Paper D . . . . .	97



# 1 Introduction

In many engineering applications it is not possible to eliminate material flaws because flaws are often generated during the production, by corrosion, or inadequate handling. These flaws can already act as critical cracks, which suddenly propagate at a critical load. This leads to sudden catastrophic failure [1], which is unacceptable in most cases. Consequently, it is crucial that engineering materials are tolerant to flaws. This damage tolerance can be quantified by the fracture toughness which corresponds to the work required to further propagate a crack [2].

A common mechanism that makes materials tougher is plastic deformation of the material, because much work goes into plastic deformation and blunting of cracks [3]. However, in some environments, plasticity cannot evolve either because one is below the brittle-ductile transition like for impact [4] or the environment leads to an embrittlement like for hydrogen tanks [5] or Polycarbonate exposed to Acetone [6].

However, toughness does not require plasticity, but can also be achieved by heterogenous features that first stop cracks and then require more energy for the initiation of new cracks. Such heterogenous features can be holes [7–10] or fibers [11], but it is also possible to use lattice [12, 13] or porous structures [14–16]. A famous example for toughening a material by adding particles is ABS [17] which consists of a brittle and stiff matrix with rubber particles with very low stiffness that act as crack stoppers. In natural materials like bones [18], nacre [19], or glass sponge skeletons [18], such toughening mechanisms can also be found at multiple length levels.



## 2 Objective of the thesis

The motivation for this work is to make materials tougher and more damage-tolerant. For example, porous materials contain many holes that can stop propagating cracks such that new crack must be initiated, which requires higher loads and leads to a high toughness [10, 11]. To understand this effect, the author investigate the effect of holes on a millimeter scale.

The objective of this thesis is to develop methods to predict crack initiation and crack propagation efficiently and accurately. The developed initiation and propagation methods are used to optimize the tensile toughness  $U_T$  of 2D specimens by adding inclusions such as holes. For this purpose, some simplifications and assumptions are made:

- Either plane stress or plane strain 2D specimens are considered.
- The material is linear elastic and isotropic.
- The crack propagation and crack initiation are brittle without plasticity.
- Dynamic effects are neglected.

Figure 2.1 illustrates the objective. The first row shows three specimens under uniaxial tensile loading. The first specimen (Fig. 2.1a) has no holes. The second specimen (Fig. 2.1b) contains three holes of various shapes. The topology of the third specimen (Fig. 2.1c) maximizes the tensile toughness  $U_T$  and is unknown. The tensile toughness  $U_T$  is computed by integrating the  $F(u)$  curve. All specimens contain an initial crack of length  $a_0$ . At a certain critical force, the initial crack starts to propagate. This event is marked as **1** and can also be seen in the force-displacement  $F(u)$  curves as the first drop in the force.

For the first specimen without any holes, the crack immediately grows through the whole specimen when the critical load is reached. The second specimen demonstrates that the tensile toughness  $U_T$  can be increased by adding holes that stop cracks. New cracks (**2**, **3**, **4**) initiate at higher applied displacements  $u$ , which leads to a higher  $U_T$ . Third specimen illustrates an optimized hole topology, such that  $U_T$  is maximized.

For the maximization of  $U_T$ , one can consider the following mechanism:

- Adding holes reduces the stiffness and subsequently the critical applied forces and the tensile toughness  $U_T$ .

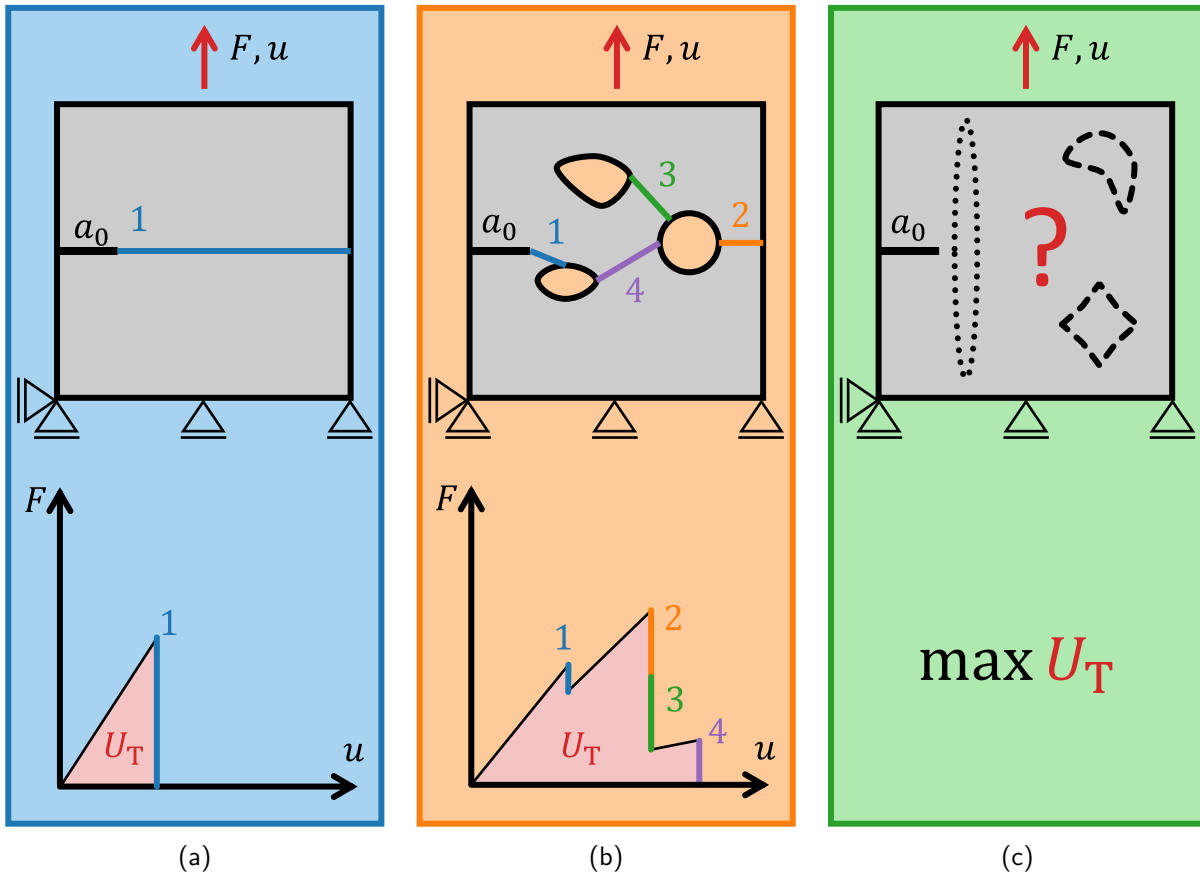


Figure 2.1: (a) Specimen with an initial crack (black) under a tensile load. The crack propagates (blue) when a critical load is reached. The tensile toughness  $U_T$  is computed from the  $F(u)$  curve. (b) Specimen with additional holes that stop cracks. New cracks (**2**, **3**, **4**) initiate at the holes. (c) The objective of this work is to find a specimen with the highest tensile toughness  $U_T$  by adding holes.

- However, adding a hole at the right position may stop a crack. Then, a new crack needs to initiate at a higher critical load. This may increase the tensile toughness  $U_T$ .
- A hole causes a notch effect: The stress increases towards the hole surface, which reduces the critical load for crack initiation. Consequently, holes should have a smooth surface with a low curvature radius to reduce the notch effect.
- Additional stress relief holes placed in a line in loading direction may reduce the notch effect, because they shield each other.
- Stress relief holes placed above a crack tip can even stop crack growth, because they reduce the critical load.

With these considerations, it is obvious that placing a long but thin vertical slit (dotted line in Fig. 2.1c) directly in front of the initial crack increase  $U_T$  significantly, because the slit stops the crack. The slit is thin and negligibly influences the stiffness in loading direction. Since the slit has a low notch effect, a high applied load is required for the



initiation of a new crack. However, if the loading situation and initial crack would rotate by  $90^\circ$ , the slit would drastically decrease  $U_T$ . Porous materials are tougher compared to the base material for all loading directions. To understand this, the author aims to increase  $U_T$  independently of the direction of the initial crack and loading direction.



### 3 Prediction of fracture

Before optimizing the tensile toughness  $U_T$ , the fracture process of a specimen must be computed. Fracture occurs when a crack breaks the specimen such that no load can be transferred anymore. To predict this cracking process, a proper representation of a specimen that contains a crack is required.

A *specimen* is a continuum body with a continuous displacement field. However, a *crack* is a discontinuity inside the specimen with a discontinuous displacement field along the crack path. At the crack tip, the discontinuous displacement field along the crack suddenly becomes continuous. This transition from discontinuous to continuous causes high stresses. The stress and strain field for elastic materials is even singular at the crack tip, which means the stresses become infinite [2]. To build a model including a crack, it is crucial to implement the discontinuous displacement field and ideally the singularity at the crack tip. Several approaches have been presented to model cracks. They can be mainly categorized into *discrete* and *smearred* models [20].

- A *Discrete* crack causes a strong discontinuity in the displacement field. A crack has two faces and both faces must be able to separate from each other. To implement this separation, one can distinguish two cases. In the first case, the crack path is known in advance and in the second case, the crack path is determined during the computation.

If the crack path is known in advance, it can be introduced as geometrical feature along which nodes are duplicated to open the crack.

If the crack path is not known in advance, the same approach is also possible, but the geometry needs to be updated and remeshed after every crack increment [20], which is slow. A more efficient approach are enriched elements [21, 22]. Elements passed by a crack are enriched with a discontinuous displacement field that allows the separation of both crack faces. The Extended Finite Element Method (X-FEM) developed from this idea [23]

- A *smearred* crack averages the discontinuity over an area and causes a weak discontinuity in the displacement field. Smearred cracks are treated as damaged areas with a low stiffness. The damaged areas have a width that influences the result significantly. Several damage models have been developed to reduce this influence. The crack band approach [24] uses an orthotropic material to only reduce the stiffness

in the direction normal to a crack. Other approaches like gradient-enriched [25] and phase-field [26, 27] models use a characteristic length to control the width of the damaged areas.

In this work, discrete cracks are introduced directly in the specimen geometry in an iterative process. In each iteration the crack propagates by an increment  $\Delta a$ . The stress singularity at the crack tip is addressed by a special crack tip mesh that is explained in the following sections. Next, criteria are needed to predict the path along which a crack grows. The criterion should correctly predict three states in the lifetime of a crack:

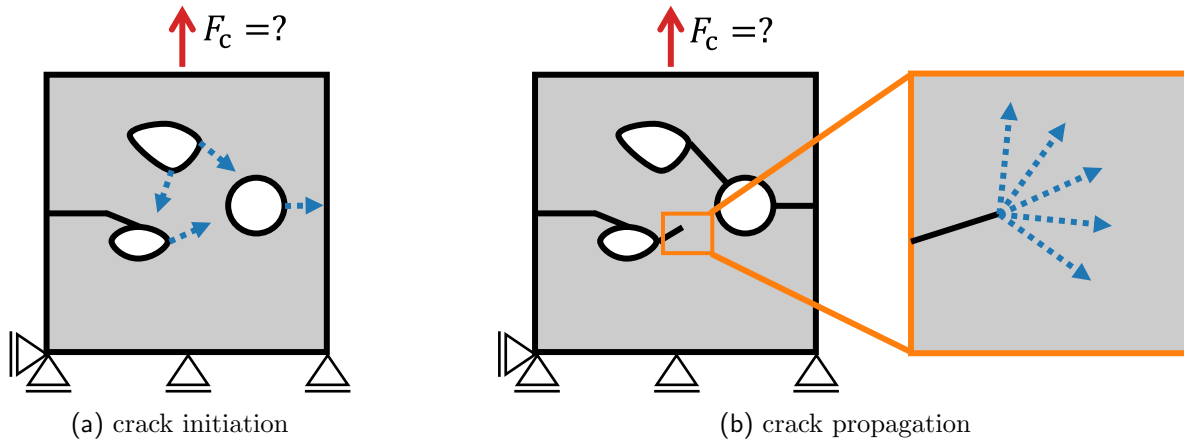


Figure 3.1: Open questions for predicting fracture in a 2D plate: (a) At which critical load  $F_c$  and where does a new crack initiate? (b) At which critical load  $F_c$  and in which direction does a crack grow?

1. *Crack initiation* is the process of developing a new crack. The new crack initiates at a surface with a finite length perpendicular to the surface. As depicted by Fig. 3.1a, several positions are possible for the crack initiation (blue arrows). A fracture criterion should predict the most critical position with the lowest critical load  $F_c$  for the crack initiation.

Under certain loading conditions like rolling contact [28], cracks may also initiate underneath the surface as "subsurface cracks". However, subsurface cracks are not considered in this work. Furthermore, the term "initiation" is not used consistently in literature and is sometimes referred to the critical load for the "initiation" of crack propagation, when an existing crack starts to grow [29].

2. *Crack propagation* presumes that a crack already exists, either because a crack initiated at a hole surface, or the specimen contains an initial crack. Crack propagation denotes the process that an existing crack grows. Figure 3.1b illustrates that the propagation direction (blue arrows) needs to be predicted as well as the critical load  $F_c$  for the crack propagation.

3. *Crack deactivation*: If an existing crack grows into a surface, it cannot grow further and is deactivated. In addition, cracks may also be deactivated if they run into a field of compressive stress.

In the following sections, three categories of criteria are explained. *Failure criteria* like the one of Mises [30] consider the stress state at each point and compare it to a critical stress state. Since the stress at a crack tip becomes infinite even for the smallest load, failure criteria always predict failure at a crack tip. This does not match with experimental results [29]. *Linear Elastic Fracture Mechanics* (LEFM) can predict the critical load of a crack, but LEFM only works for existing crack and is not able to predict crack initiation from surfaces. *Finite Fracture Mechanics* works on both, crack tips and smooth notches [31].

### 3.1 Failure criteria

Failure criteria consider the computed local stress (or strain) tensor for each point in a specimen. Furthermore, a failure surface is defined that depends on experimentally obtained material parameters like the strength or the yield stress. Failure is predicted if the local stress tensor reaches the failure surface.

Figure 3.2 illustrates this procedure. The left side shows the computed stress output of a specimen with a heart-shaped hole subjected to a load  $F$ . The local (in-plane) stress tensor is computed at two points (A) and (B).

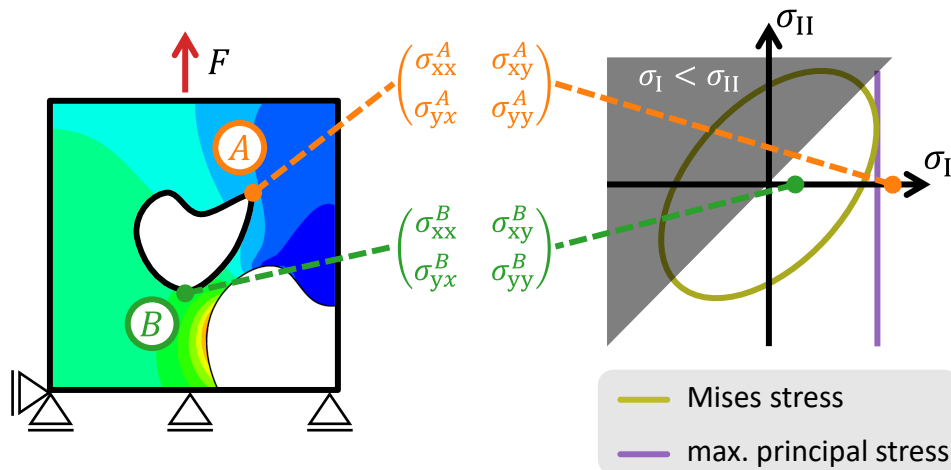


Figure 3.2: The stress tensors at two points (A) and (B) are used to check if failure occurs. For this, failure surfaces are plotted on the right in the principal stress coordinate system for the Mises criterion and the (in-plane) max. principal stress criterion. Point (B) is inside both failure surfaces and will not fail. At point (A) failure occurs according to both criteria. Note, that  $\sigma_I$  is always greater or equal to  $\sigma_{II}$ .

The right side of Fig. 3.2 shows the failure surface for the Mises [30] and max. (in-plane) principal stress criterion in the principal component coordinate system  $(\sigma_I, \sigma_{II})$ . To show

whether failure occurs, the computed local stress tensors are also plotted in the principal component coordinate system. Point (B) is inside the failure surfaces and will not fail contrary to point (A) which exceeds the failure surfaces.

As can be seen in Fig. 3.2, the failure surfaces of the Mises criterion and the max. principal stress criterion have different shapes and one has to decide which failure criterion is better suited for the problem. The Mises criterion is applicable especially for ductile materials. In this case, failure is defined when the yield stress is reached and plastic deformation occurs [32]. This work, however, deals with cracking processes that occur in brittle materials, where the max. principal stress criterion is better suited.

Still, the max. principal stress criterion has the problem, that it cannot be applied in the presence of a crack, due to the stress singularity at the crack tip. Nevertheless, in some cases the max. principal stress criterion can be used to efficiently find the most critical position for crack initiation along a surface. For example, Li and Leguillon [33] apply a watershed flooding algorithm to the max. principal stress,  $\sigma_I$ , around a hole and thereby find critical positions. This is an efficient approach, because only one FEM simulation is required to compute  $\sigma_I$ . In the next step, Li and Leguillon apply a more complicated criterion to these critical positions. Paper D [34], presented in this thesis, also uses  $\sigma_I$  to find the critical positions on hole surfaces.

## 3.2 Linear Elastic Fracture Mechanics

As mentioned in the previous section, failure criteria are not applicable in the presence of cracks, because of the stress singularity at the crack tip. The fracture mechanics concept, however, is designed exactly for this case. According to some authors [2], already Leonardo Da Vinci investigated some aspects of fracture in 1500, when he tested the strength of wires of various length and found that the strength decreases as the length of the wire increases [35]. Since the probability of flaws (micro-cracks) in the wire increases with the volume, it can be concluded there is a connection between material flaws and the strength of the wire. The following sections give an overview on how to use Linear Elastic Fracture Mechanics (LEFM) for predicting the direction of crack propagation and the critical load.

### 3.2.1 Griffith's G-concept

Today's view on fracture mechanics dates back to Griffith [36]. In 1921, he treated cracks as very flat elliptical holes, for which Inglis [37] provided analytical equations for the stress field. Thereby, Griffith was able to formulate an energy-based criterion according to which a crack propagates if the change in strain energy,  $\Pi$ , is higher than the work,  $W_s$ , required to create two crack faces. This occurs if a critical displacement  $u_c$  is applied to a specimen. In relation to the newly created crack area  $dA$  the criterion can be written as:

$$-\left(\frac{d\Pi}{dA}\right)_{u_c} = \frac{dW_s}{dA}. \quad (3.1)$$

Like for the failure criteria, Griffith's criterion can be split into a left and a right side. The left side is called (differential) energy release rate [38]

$$G := -\left(\frac{d\Pi}{dA}\right)_{u_c} \quad (3.2)$$

which is computed e.g. using FEM computations and which depends on the applied load, the crack length, and the specimen geometry. The right side is denoted as the critical energy release rate

$$G_c := \frac{dW_s}{dA} \quad (3.3)$$

which is a material parameter determined experimentally.

### 3.2.2 Irwin's K-concept

An evaluation of Griffith's G-concept requires the energy release rate which cannot be computed analytically for most cases. Irwin's K-concept instead quantifies the stress singularity at the crack tip, as illustrated by Fig. 3.3. Westergaard [39], Irwin [40], and Williams [41] found that for brittle elastic materials the stress in front of a crack tip

$$\sigma(r) = k \cdot f_{\text{shape}} \cdot \frac{1}{\sqrt{r}} + \text{higher order terms} \quad (3.4)$$

is related to a load factor  $k$ , a geometrical factor  $f_{\text{shape}}$ , and the distance to the crack tip  $r$ . Near the crack tip, higher order terms are negligible compared to the singularity  $1/\sqrt{r}$  that approaches infinity for  $r \rightarrow 0$ . It is a convention to define the load factor as so-called stress intensity factor  $K = k/\sqrt{2\pi}$  [2]. The fracture toughness  $K_c$  is then the critical stress intensity factor at which a crack propagates.

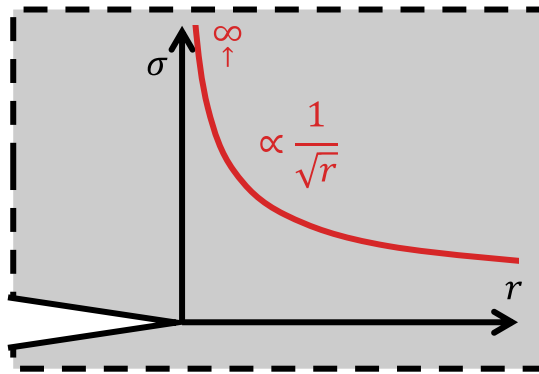


Figure 3.3: The stress  $\sigma$  in front of a crack tip is proportional to  $1/\sqrt{r}$  and approaches  $\infty$  when  $r \rightarrow 0$ .

The K-concept allows to split the load at the crack tip into three modes, depicted by

Fig. 3.4. For each mode, a stress intensity factor  $K_I$ ,  $K_{II}$ ,  $K_{III}$  can be defined. However, in this work only 2D plates are considered, hence only mode I and mode II loadings are possible.

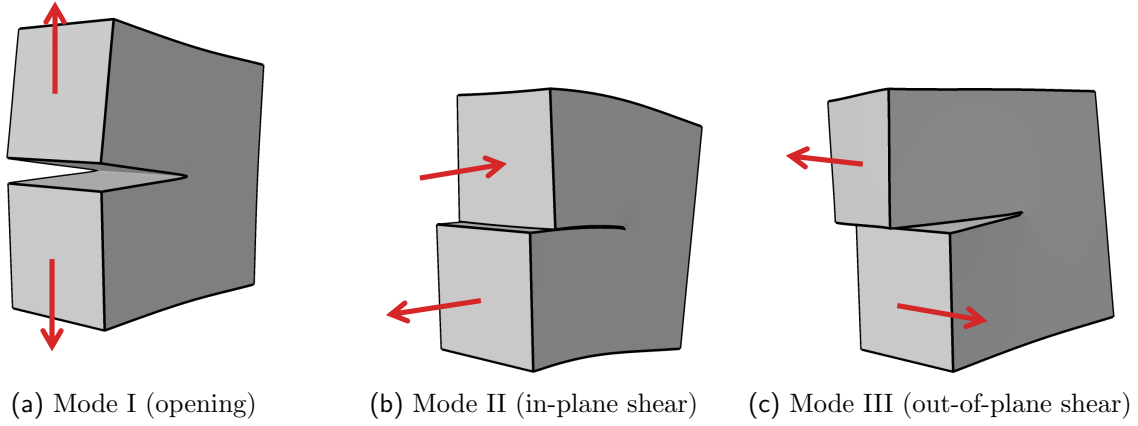


Figure 3.4: Three loading modes at a crack tip.

According to the K-concept, a crack under a pure mode I loading propagates if

$$K_I = K_{Ic} \quad (3.5)$$

is fulfilled. The stress intensity factor  $K_I$  is computed e.g. using FEM computations, whereas the fracture toughness  $K_{Ic}$  is determined experimentally. Analogously, for pure mode II loadings the mode II stress intensity factor  $K_{II}$  is compared to the mode II fracture toughness  $K_{IIc}$  [2]. However, this simple criterion does not work for mixed mode loadings and more complicated criteria are required in this case [42].

Irwin [40] showed, that the K-concept is equivalent to the G-concept for a plate under mode I loading. Assuming a linear elastic material, the fracture toughness  $K_{Ic}$  can be converted to the critical energy release rate

$$G_c = \begin{cases} K_{Ic}^2 \cdot \frac{1}{E} & \text{(plane stress)} \\ K_{Ic}^2 \cdot \frac{1-\nu^2}{E} & \text{(plane strain)} \end{cases} \quad (3.6)$$

with the Young's modulus  $E$  and the Poisson's ratio  $\nu$ . The formulation depends on whether there is a plane stress or plane strain state.

### 3.2.3 Mixed mode I-II

Under a pure mode I loading, a crack propagates in a straight line. However, if there is a mixed mode I and mode II loading, the crack propagation direction shows a kink, as depicted in Figure 3.5a. The propagation direction  $\alpha$  is unknown and needs to be predicted by a mixed mode criterion. [43]

Erdogan and Sih [44] proposed the stress-based Maximum Tangential Stress (MTS)



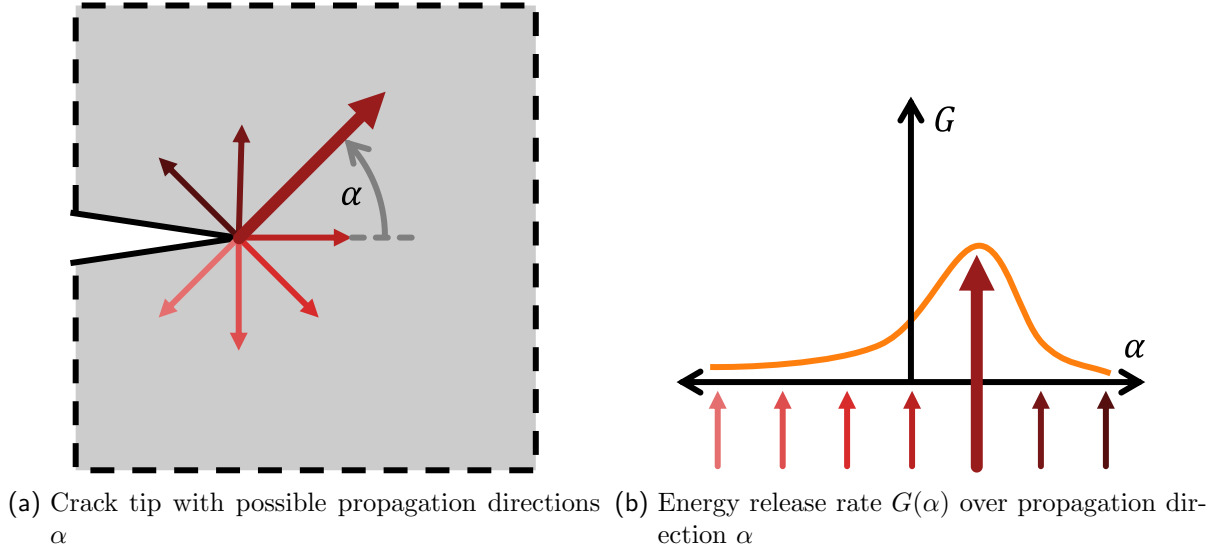


Figure 3.5: The crack grows in the direction of the maximum energy release rate. The corresponding crack path is plotted as large bold arrow.

criterion. According to MTS, a crack propagates in the direction perpendicular to the largest tension around the crack tip. The direction can be computed for a plane strain plate from the stress intensity factors  $K_I$  and  $K_{II}$ :

$$\cos\left(\frac{\alpha}{2}\right) \cdot [K_I \cdot \sin(\alpha) + K_{II} \cdot (3 \cdot \cos(\alpha) - 1)] = 0. \quad (3.7)$$

A few years later, Sih [45] developed another energy-based criterion, called Minimum Strain Energy Density (MSED). This criterion considers the strain energy density

$$\psi := \frac{d\Pi}{dV} \quad (3.8)$$

in front of the crack tip as the strain energy  $\Pi$  per volume  $V$ . According to MSED, a crack propagates in the direction of the minimum  $\psi$ . Boulenouar [46] used a ring of FEM elements around a crack tip and computes  $\psi$  for each element. He showed that  $\psi$  has several minima in the ring of elements and the global minimum is not necessarily the physically meaningful propagation direction.

In this thesis, the focus lies on another energy-based criterion, called Maximum Energy Release Rate (MERR) [47]. It captures the idea of Griffith, according to which cracks grow in the direction of the maximum change in strain energy. However, for a long time it was too complicated to compute the direction of the maximum energy release rate. Only the development of the J-Integral in 1968 and 1973 by Budiansky and Rice [48, 49] allowed an efficient computation of the energy release rate  $G(\alpha)$  for a crack propagating in the direction  $\alpha$ . Using this, the max. energy release rate and the corresponding propagation direction can be computed according to MERR, as illustrated by Fig. 3.5b.

### 3.2.4 Computation of the energy release rate

The MERR criterion for crack propagation needs the energy release rate,  $G$ . In this section, three approaches for the computation of  $G$  are presented. The *Full FEM* approach is the most versatile, but computationally most expensive. An efficient approximation of  $G$  is the *J-Integral* [48, 49]. *Configurational forces* are similar to the J-Integral, but are easier to compute in FEM simulations.

#### 3.2.4.1 Full FEM approach for crack propagation

The Full FEM approach uses two separate Finite Element Method (FEM) simulations with the initial and an extended crack. It approximates the energy release rate  $G = -(d\Pi/dA)_{u_c}$  by a numerical differentiation of the strain energy  $\Pi$  with respect to the crack length  $a$ :

$$G \approx - \left( \frac{\Delta\Pi}{\Delta A} \right)_{u_c} = - \left( \frac{\Pi(a + \Delta a) - \Pi(a)}{b \cdot \Delta a} \right)_{u_c} \quad (3.9)$$

Since this thesis considers 2D plates, the crack area  $A$  can be decomposed in the plate thickness  $b$  and the crack length  $a$ . The crack increment  $\Delta a$  should be a small value, because the numerical differentiation converges to the exact solution  $(\Delta\Pi/\Delta A) \rightarrow (d\Pi/dA)$  for small increments  $\Delta a \rightarrow 0$ .

Figure 3.6 illustrates the implementation of the Full FEM approach. The specimen, depicted in Fig. 3.6a, contains a crack of length  $a$  and is either subjected to a critical force  $F_c$  or to a critical displacement  $u_c$ . The specimen is meshed, such that there is an element edge in the direction of a trial crack propagation  $\mathbf{q}$ . This allows to open the element edge in the next step without remeshing the specimen (Fig. 3.6b). Remeshing would lead to additional numerical errors. If  $\mathbf{q}$  is not known in advance, the procedure has to be repeated for several trial directions. To open the element edge, the turquoise nodes are duplicated and the crack propagates by  $\Delta a$ . It is important that the specimen is not remeshed when the nodes are duplicated, because this would add small numerical errors to  $\Pi(a)$  and  $\Pi(a + \Delta a)$ , which would significantly influence  $\Pi(a + \Delta a) - \Pi(a)$  and consequently  $G$ .

If a crack propagates, one has to decide if a displacement-controlled or a force-controlled approach should be used. The displacement-controlled approach holds the critical displacement  $u_c$  constant. The corresponding  $F(u)$  curves are plotted in Fig. 3.6c and the strain energies  $\Pi_1$  and  $\Pi_2$  can be computed. The change in the strain energies  $(\Delta\Pi)_{u_c} = \Pi_2 - \Pi_1$  with a fixed  $u_c$  can be inserted into Equation 3.9, which yields  $G$ .

For the force-controlled approach, the computation of  $G$  is more complicated. The critical force  $F_c$  is held constant, as depicted by Fig. 3.6d. Since a propagating crack reduces the stiffness, the applied displacement  $u$  becomes larger for a fixed  $F_c$ . This requires additional external work  $\Delta\Pi_{\text{ext}}$ , which must be considered. Figure 3.6e shows this external work  $\Delta\Pi_{\text{ext}}$  that must be subtracted to obtain  $(\Delta\Pi)_{u_c} = (\Pi_2 - \Pi_1) - \Delta\Pi_{\text{ext}}$ .

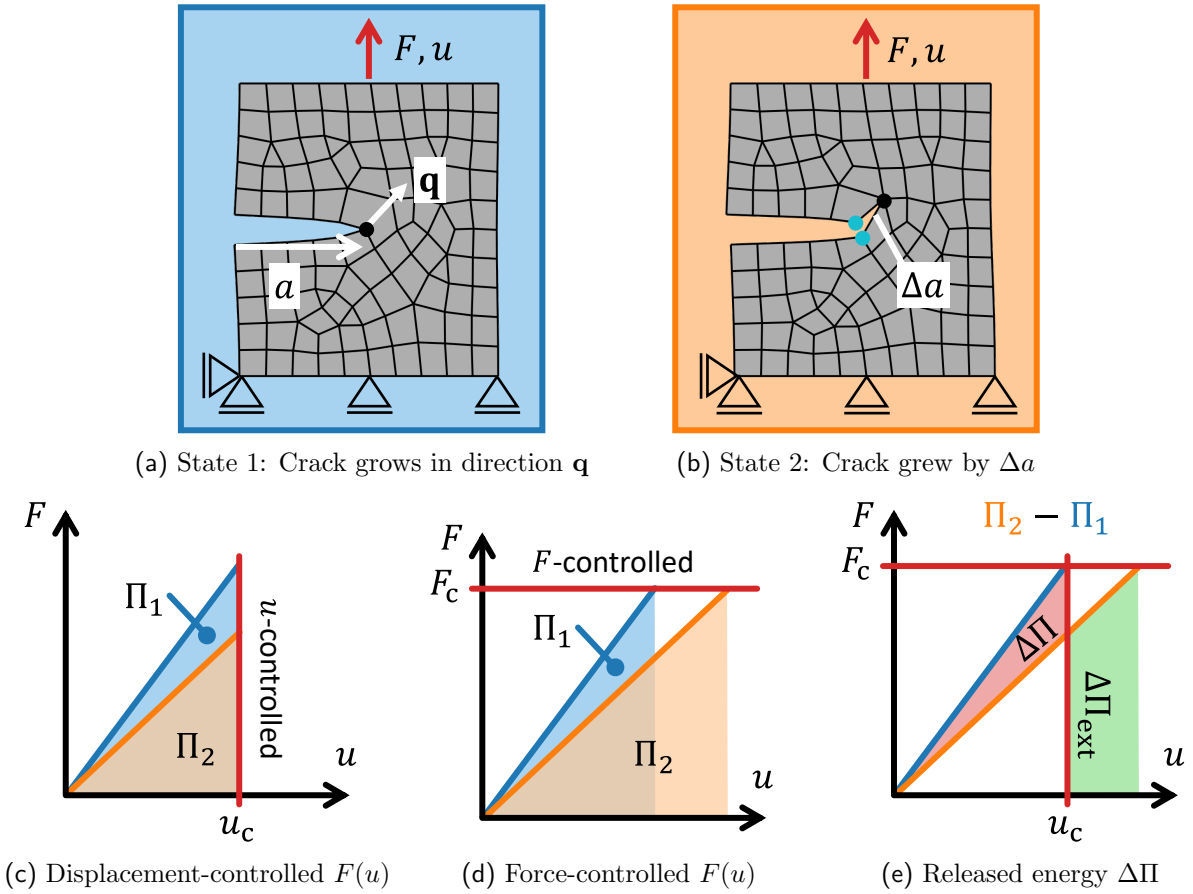


Figure 3.6: Energy released by a growing crack: A crack growing in direction  $\mathbf{q}$  by  $\Delta a$  changes the strain energy from (a)  $\Pi_1$  to (b)  $\Pi_2$ . The strain energy  $\Pi_2$  depends on whether (c) the critical displacement  $u_c$  or (d) the critical force  $F_c$  is held constant. (c) The energy release  $\Delta\Pi$  for the displacement-controlled approach  $\Delta\Pi = \Pi_2 - \Pi_1$  and for the force-controlled approach is  $\Delta\Pi = \Pi_2 - \Pi_1 - \Delta\Pi_{\text{ext}}$ , where the latter considers the additional external work  $\Delta\Pi_{\text{ext}}$ .

Next,  $G$  can be computed by inserting  $(\Delta\Pi)_{u_c}$  into Equation 3.9.

A drawback of the full FEM approach is that two FEM simulations are required to compute the energy release rate  $G$  for one propagation direction  $\mathbf{q}$ . As mentioned in the previous section, a crack propagates in the direction of the maximum energy release rate. To find this direction,  $G$  has to be computed for several trial directions and every time a FEM simulation is required. Consequently, the full FEM approach needs many FEM simulations and is too inefficient for most applications.

### 3.2.4.2 J-Integral

Sine the full FEM approach can be computationally expensive for obtaining the crack propagation direction, an alternative computation of  $G$  is required. The path-independent J-Integral, presented by Rice [48] in 1968, efficiently approximates  $G$  for a crack that propagates in a straight line. This approach was extended by Budiansky and Rice [49] in 1973 to the vectorial J-Integral

$$\mathbf{J} = \begin{pmatrix} J_x \\ J_y \end{pmatrix} = \oint_{\Gamma} \left( \psi \cdot n_x - \mathbf{t} \cdot d\mathbf{u}/dx \right) \quad (3.10)$$

which considers a closed path  $\Gamma$  that includes both crack faces  $\Gamma_+$  and  $\Gamma_-$ , as depicted in Fig. 3.7. This vectorial J-Integral approximates also the propagation direction. The first term contains the strain energy density  $\psi$  and the normal vector  $\mathbf{n}$  of  $\Gamma$ . It corresponds to the change of the strain energy inside the region  $\Omega$  that is enclosed by  $\Gamma$ . The second term corresponds to the external work performed on  $\Omega$  due to a load redistribution caused by a propagating crack. This term evaluates the gradient of the displacement vector  $\mathbf{u}$  and the traction  $\mathbf{t}$  at the path  $\Gamma$ .

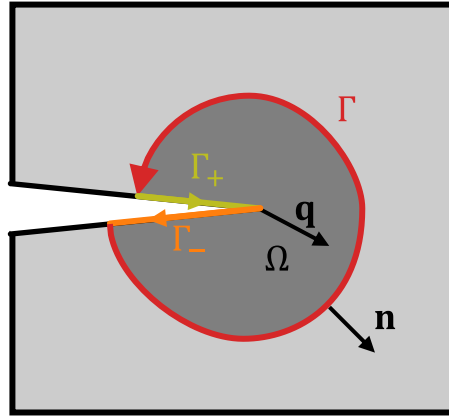


Figure 3.7: Domains of a crack: The region  $\Omega$  surrounds the crack tip and is mainly controlled by the stress singularity at the crack tip. The vector  $\mathbf{q}$  points in the crack propagation direction. The outer contour of  $\Omega$  is  $\Gamma$ . Note, that this also includes the two crack faces  $\Gamma_+$  and  $\Gamma_-$ . The normal vector of  $\Gamma$  is  $\vec{n}$ .

Using the vectorial J-Integral, the energy release rate  $G \approx b \cdot \mathbf{J} \cdot \mathbf{q}'$  can be approximated by a projection on an arbitrary crack propagation direction  $\mathbf{q}'$  and by multiplying with the thickness of the plate  $b$  [50]. The direction  $\mathbf{q}'$  is a unit vector and can be varied to maximize  $G$  according to the MERR criterion. The projection  $\mathbf{J} \cdot \mathbf{q}'$  and subsequently  $G$  becomes a maximum if  $\mathbf{q}'$  is parallel to  $\mathbf{J}$ . Consequently, the propagation direction  $\mathbf{q}$  according to MERR is  $\mathbf{q} = \mathbf{J} / \sqrt{J_x^2 + J_y^2}$ .

The J-Integral is path-independent for a crack in an infinite plate. This means that it does not matter if the path  $\Gamma$  surrounds just a small region around the crack tip or a larger region around the whole crack. This is valid for an infinite plate containing one crack. However, the region needs to be small enough to enclose only one crack at a time, otherwise it would no longer be possible to compute  $G$  for each crack independently. To prevent enclosing more cracks, it is common to compute  $\mathbf{J}$  just around the a small region at the crack tip and pay special attention to the FEM mesh in this region.

Referring to Fig. 3.8, the J-Integral can be evaluated for any of the red contours. However, for the inner-most contour, the J-Integral may be underestimated, because the FEM computation with the crack tip mesh is still just an approximation. A common

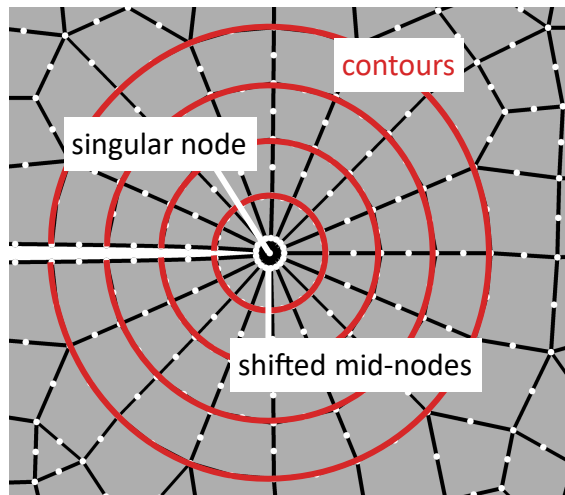


Figure 3.8: The crack tip mesh accounts for the  $1/\sqrt{r}$  singularity in front of a crack tip. Quadratic rectangular elements are placed in circles around the crack tip. Each circle corresponds to a contour. In the inner ring, the mid-nodes are shifted 25% towards the crack tip and the inner nodes of the rectangular elements are collapsed to a single position such that a singular stress field can be accounted for and the quadrilateral elements look like triangles, respectively.

approach to decide which contour to use, is to evaluate all contours and take the first contour where the corresponding J-Integral stays approximately constant.

Although, the J-Integral efficiently approximates  $G$ , it has a few drawbacks that are described in the following.

**Similarity assumption** The J-Integral assumes a similarity between the current crack and the crack after an infinitesimal propagation of  $da$ . Figure 3.9 illustrates this similarity assumption. The energy release rate  $G$  is related to the gradient of the strain energy with respect to a crack propagation of  $da$ . However, the J-Integral computes the gradient with respect to  $dX$  instead to  $da$ . As depicted by the blue areas in Fig. 3.9, a shift of  $-dX$  corresponds to a moving viewpoint. The orange areas show an actual crack propagation of  $da$  with a fixed viewpoint.

Now, assume you are standing at the viewpoint and look at a region near the crack tip. If the shift of  $-dX$  leads to the same result as a propagation of  $da$ , the similarity is fulfilled and the J-Integral is a valid approximation for  $G$ . However, if a feature like a hole, a boundary condition, etc. exists in the vicinity of the crack tip, the feature is shifted together with  $-dX$ , but stays fixed in the case of a crack propagation of  $da$ . In this case, the J-Integral is not valid.

As mentioned previously, the J-Integral can be evaluated for various contours. If one uses a smaller contour, that encloses a smaller region, the probability of additional features in the observed region decreases. Consequently, the similarity assumption is more likely to be fulfilled for smaller contours.

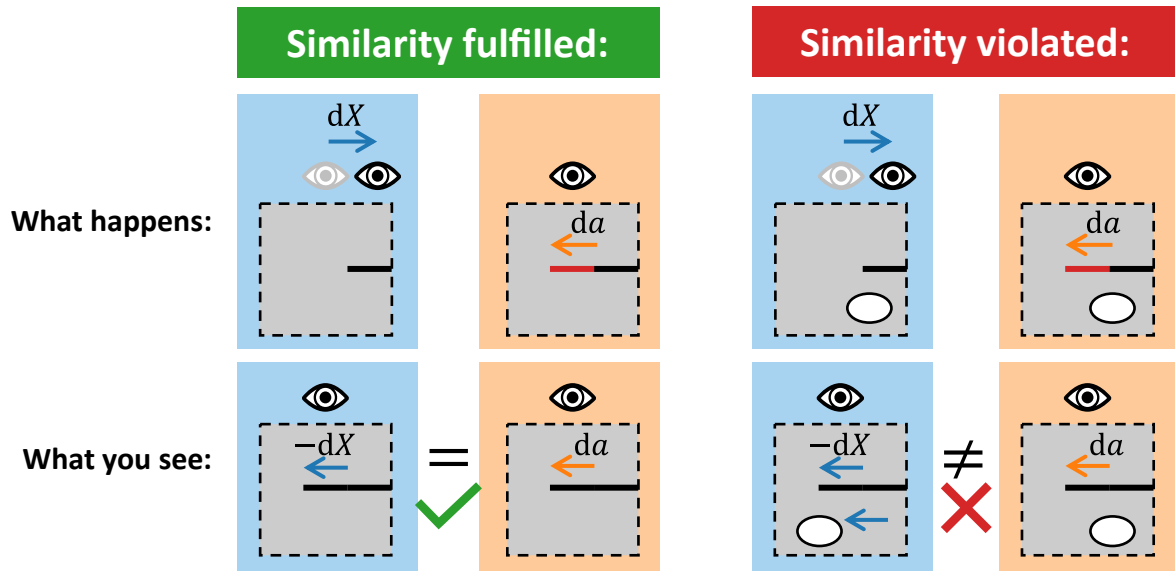


Figure 3.9: Similarity between changing the viewpoint by  $-dX$  and growing a crack by  $da$ . The viewpoint, depicted by an eye, observes a region near the crack tip. The top row shows what really happens. In the blue columns, the viewpoint moves by  $dX$ . In the orange columns, the viewpoint is held constant, but the crack grows by  $da$ . The bottom row shows what you see from the viewpoint. On the left side, moving the viewpoint is equivalent to crack growth. On the right side, there is a void included, which moves in common with the viewpoint, but not with a growing crack. Consequently, the similarity is not fulfilled.

**Crack kinking** The similarity assumption is also not fulfilled if a crack has a kink, as depicted by Fig. 3.10. The blue contour in Fig. 3.10a is used for the evaluation of the J-Integral. The red arrow points in the direction of the crack propagation direction  $\mathbf{q}$ . If the similarity assumption would be fulfilled, the cracks shown in Fig. 3.10b and Fig. 3.10c would look the same. Obviously, this is not the case.

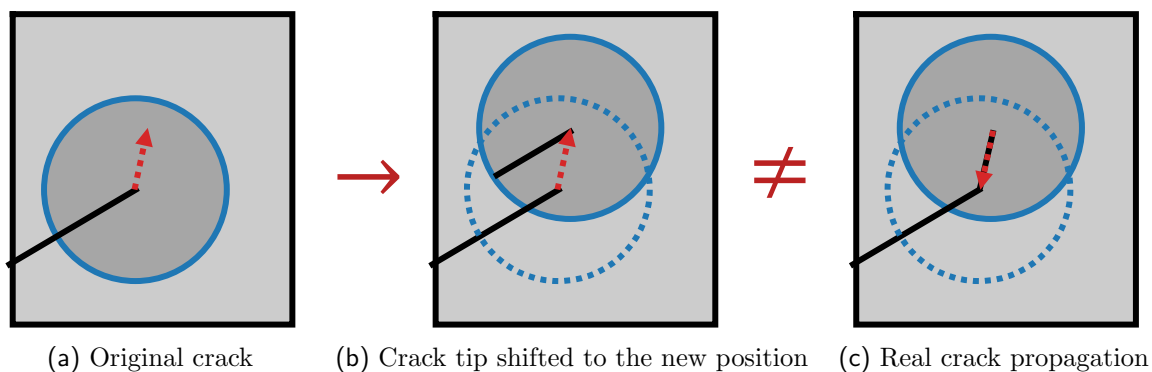


Figure 3.10: If the crack propagation direction has a kink, shifting the crack tip to the new position is inaccurate. (a) The crack propagation direction (red arrow) deviates from previous direction. (b) The crack tip is shifted to the new position. This does not equal (c) the real crack propagation.

Frankl et al. [51] suggested a correction method to account for crack kinking, by which the propagation direction  $\mathbf{q}'$  is iteratively optimized in several FEM simulations. In each FEM simulation, a trial crack increment in a direction  $\mathbf{q}'_0$  is introduced. Based on the trial

crack increment, the vectorial J-Integral is evaluated and  $\mathbf{q}'_i$  is updated. The optimization stops, when  $\mathbf{q}'$  is aligned with the actual propagation direction within a tolerance angle as depicted by Fig. 3.10c.

However, this approach requires additional FEM simulations, and subsequently more computation time. Crack kinking occurs in mode II or mixed mode I/II loadings. Judt and Ricoeur [50] showed, that the J-Integral is valid for a small mode II contribution with  $K_{II}/K_I < 0.1$ . For higher mode II contributions, a correction scheme is required.

**J-Integral based on FEM results** The evaluation of the J-Integral directly from FEM results is cumbersome. The FEM approach provides a continuous displacement field  $\mathbf{u}$ , but the gradient of  $\mathbf{u}$  is discontinuous. Since the traction  $\mathbf{t}$  is related to the gradient of  $\mathbf{u}$ , it is discontinuous at element edges. However, the J-Integral uses Green's theorem that  $\mathbf{t}$  is continuous. Consequently, alternative approaches are required to compute the J-Integral from FEM results. The software Abaqus [52] uses the virtual crack extension method [53]. Another alternative are configurational forces [54].

### 3.2.4.3 Configurational forces

As mentioned in the previous section, the evaluation of the J-Integral based on FEM results is difficult. Configurational forces, however, can be computed easier from FEM results and are identical to the J-Integral for elastic bodies [48]. They base upon Eshelby's energy-momentum tensor [55]

$$\Sigma^{(\text{mbf})} := \psi \cdot \mathbf{I} - \mathbf{F}^\top \cdot \mathbf{P} \quad (3.11)$$

with the strain energy density  $\psi$ , the identity matrix  $\mathbf{I}$ , the deformation gradient tensor  $\mathbf{F}$ , and the first Piola-Kirchhoff stress tensor  $\mathbf{P}$ . Eshelby showed that the divergence of the energy momentum tensor

$$\frac{d\Sigma}{d\mathbf{x}} = \left( \frac{\partial\psi}{\partial\mathbf{x}} \right)_{\text{expl.}} \quad (3.12)$$

equals the explicit gradient of the strain energy density  $\psi$  with respect to  $\mathbf{x}$ . The explicit gradient is basically a partial derivative for  $\mathbf{x}$  with the deformation gradient held constant. Equation 3.12 can be used to approximate the energy release rate per thickness:

$$\frac{G}{b} = - \left( \frac{d\Pi}{da} \right)_{u_c} = - \int_{\Omega} \left( \frac{d\psi}{da} \right)_{u_c} dV \approx \mathbf{q}' \cdot \int_{\Omega} \left( \frac{\partial\psi}{\partial\mathbf{x}} \right)_{\text{expl.}} dV = \mathbf{q}' \cdot \int_{\Omega} \frac{d\Sigma}{d\mathbf{x}} dV. \quad (3.13)$$

The energy release rate  $G$  is related to the change in the strain energy  $\Pi$  with respect to the crack area  $A = b \cdot a$  at a load  $u_c$ . The strain energy  $\Pi$  can also be expressed as integral of the strain energy density  $\psi = d\Pi/dV$ , which is the derivative of the strain energy with

respect to the volume  $V$ . Furthermore, similarity between a crack propagation of  $da$  and a shift of the viewpoint of  $-d\mathbf{x}$  in the direction of the crack propagation  $\mathbf{q}'$  can be assumed, as explained in the previous section by Fig. 3.9. This allows to approximate  $G$  in terms of the divergence of the energy momentum tensor  $\Sigma$  and the crack propagation direction  $\mathbf{q}'$ .

For the evaluation of  $G$  from FEM results, it is convenient to reformulate Equation 3.13 such that the element shape functions  $h_i$  is derived instead of  $\Sigma$ . This approach is described by Mueller and Maugin [56], who defined nodal configurational forces

$$\mathbf{g}_{\text{nodal},i}^{(\text{mbf})} = \underbrace{\int_{\Gamma} (\Sigma^{(\text{mbf})} \mathbf{n}) \cdot h_i dS}_{\mathbf{g}_{\text{nodal},i}^{(\text{mbf},S)}} - \underbrace{\int_{\Omega} \Sigma^{(\text{mbf})} : \frac{\partial h_i}{\partial \mathbf{x}} dV}_{\mathbf{g}_{\text{nodal},i}^{(\text{mbf},V)}} \quad (3.14)$$

that are computed for each node in a crack tip dominated region  $\Omega$  with an outer contour  $\Gamma$ , as shown in Fig. 3.11. The energy release rate  $G$  is then approximated by the J-Integral

$$\mathbf{J}_{\text{CF}} = - \sum_{i \in \Omega} \mathbf{g}_{\text{nodal},i}^{(\text{mbf},V)}, \quad (3.15)$$

which points in the opposite direction of the summed up nodal configurational forces. Note, that commonly only the volume nodal configurational forces  $\mathbf{g}_{\text{nodal},i}^{(\text{mbf},V)}$  are summed up, but not the surface nodal configurational forces  $\mathbf{g}_{\text{nodal},i}^{(\text{mbf},S)}$ , because the computation of  $\mathbf{g}_{\text{nodal},i}^{(\text{mbf},S)}$  from FEM results is tedious and they are negligible for mode I loadings with only a small mode II contribution [57]. This simplification is valid, because  $\mathbf{g}_{\text{nodal},i}^{(\text{mbf},S)}$  vanishes inside the body and only contributions from the crack faces  $\Gamma_+$  and  $\Gamma_-$  remain. For mode I loadings, the crack faces are symmetric and cancel out, but for mixed mode loadings, the crack faces are not symmetric and there may be a contribution of the crack faces  $\Gamma_+$  and  $\Gamma_-$  to  $\mathbf{g}_{\text{nodal},i}^{(\text{mbf},S)}$ . Schmitz and Ricoeur [57] provide a correction scheme to account for the influence of  $\Gamma_+$  and  $\Gamma_-$  for mixed mode loadings.

As depicted by Fig. 3.11, the nodal configurational force right at the crack tip has the largest contribution of  $J_{\text{CF}}$ , but there are also some smaller nodal configurational forces near the crack tip. These configurational forces are called spurious nodal configurational forces and arise from jumps in the discontinuous stress field. Denzer et al. [58] showed, that it is important to not only consider the nodal configurational force at the crack tip, but to sum up all nodal configurational forces in the crack-dominated region  $\Omega$ .

Paper B [59] describes an implementation of the configurational forces. This implementation is used in Paper D [34] to predict crack propagation.



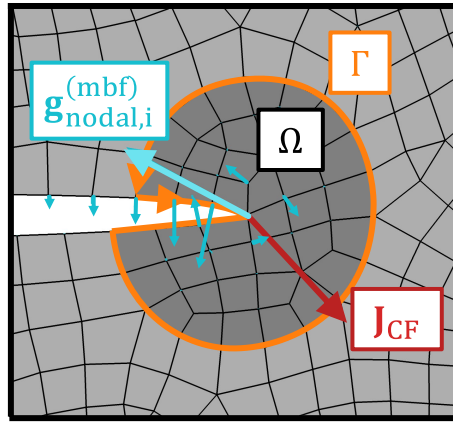


Figure 3.11: Nodal configurational forces  $\mathbf{g}_{\text{nodal},i}^{(\text{mbf},V)}$  near a crack tip. The vectorial energy release rate  $\mathbf{J}_{\text{CF}}$  is the negative sum of all nodal configurational forces in the crack dominated dark gray region. Note, that the actual vectorial energy release rate points in the opposite direction.

### 3.3 Finite Fracture Mechanics

Fracture mechanics is useful for analyzing cracks. Stress-based failure criteria can predict fracture at holes. But neither fracture mechanics nor stress-based failure criteria are suitable for both, cracks and holes. Finite Fracture Mechanics (FFM), however, can be applied to cracks as well as to holes.

FFM goes back to Hashin [60] who investigated composites and observed that cracks initiate suddenly in a ply with a finite length and "it is not possible or of interest to follow the history of their development". Based on this observation, Hashin suggested FFM as a new framework that considers crack formations of finite lengths.

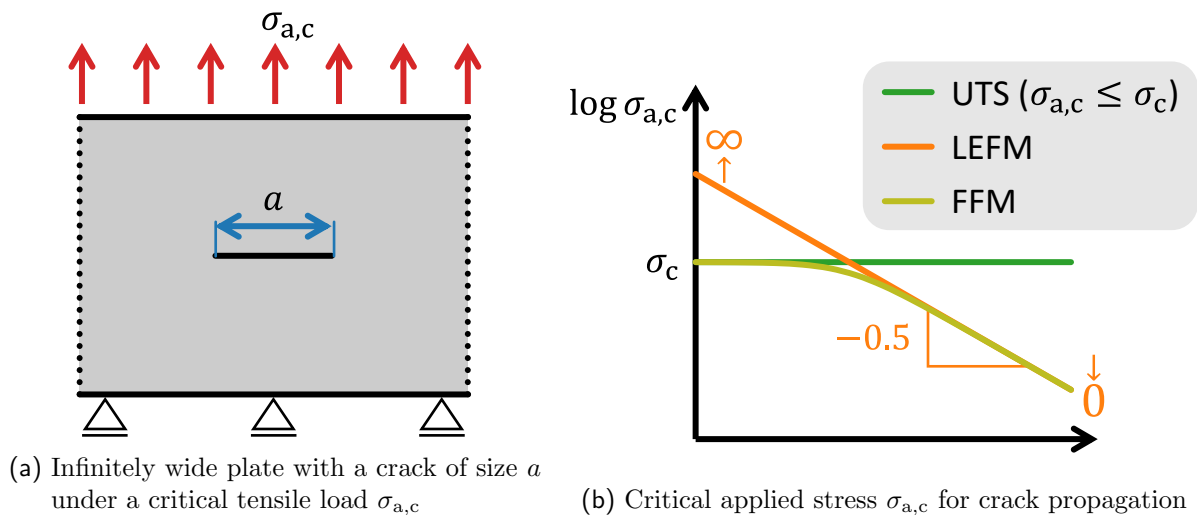


Figure 3.12: Critical applied stress  $\sigma_{a,c}$  for (a) the infinitely wide plate predicted by (b) three criteria: Ultimate tensile stress (UTS), Linear Elastic Fracture Mechanics (LEFM), and Finite Fracture Mechanics (FFM).

Later, Taylor [61] reinvented FFM as an extension of LEFM, which cannot handle very small cracks. Taylor investigated such small cracks in an infinite plate loaded by a uniaxial

tensile load  $\sigma_a$ , as depicted in Fig. 3.12a and found that according to LEFM, the critical load is higher than the material's strength. This is because LEFM relates the critical applied load  $\sigma_a \propto K_{Ic}/\sqrt{a}$  to the crack length  $a$  and the fracture toughness  $K_{Ic}$ . For small cracks  $a \rightarrow 0$ , the critical applied load  $\sigma_{a,c} \rightarrow \infty$  becomes infinite, as depicted in Fig. 3.12b. Such an infinite critical load is unphysical, especially because a plate without a crack would already break according to the Ultimate Tensile Stress (UTS) criterion if the applied stress  $\sigma_{a,c}$  reaches the material strength  $\sigma_c$ . Taylor showed that FFM fits LEFM for long cracks and UTS for short cracks and thereby leads to physical results.

There are several FFM approaches and they can be divided into two groups depending on how the size of the finite crack formation  $a_c$  is handled.

- Criteria based on the Theory of Critical Distances (TCD) [62] assume that  $a_c$  is a material parameter that relates to the material's characteristic length  $l_{ch}$ .
- Coupled Criteria (CC) [31, 63, 64] require both a stress- and an energy-based criterion to be fulfilled simultaneously. In this case,  $a_c$  is not a material parameter, but a result.

### 3.3.1 Theory of Critical Distances

In 1999, Taylor [65] investigated the fatigue limit of specimens with cracks and notches. He found that the fatigue limit can be predicted accurately if stress values are evaluated in a critical distance in front of a notch or a crack tip or if the stress is averaged over a critical distance. This critical distance is often related to the grain size in metals and ceramics [66]. Commonly, the critical distance is proportional to the material's characteristic length [62]

$$l_{ch} = \frac{1}{\pi} \cdot \left( \frac{K_{Ic}}{\sigma_c} \right)^2, \quad (3.16)$$

which is a constant material parameter that is computed from the material's fracture toughness  $K_{Ic}$  and tensile strength  $\sigma_c$ .

Taylor proposed a sequence of Theory of Critical Distances (TCD) criteria like the Point Method (PM) and the Line Method (LM). Figure 3.13 illustrates those criteria for a plate with a U-notch under a uniaxial load  $F$ . A trial crack path  $a$  is defined from the notch surface, as depicted in Fig. 3.13a. In this thesis, the crack path is always assumed to be normal to the notch surface.

The PM (Fig. 3.13b) uses the stress value  $\sigma_{PM} = \sigma(l_{ch}/2)$  at a distance of  $l_{ch}/2$  in front of a notch or a crack tip. Contrary, the LM (Fig. 3.13c) averages the stress along a line

$$\sigma_{LM} = \frac{1}{2 l_{ch}} \int_0^{2 l_{ch}} \sigma(a) da \quad (3.17)$$

from the surface to a distance of  $2 \cdot l_{ch}$ . Note, that the distance for PM is a quarter of

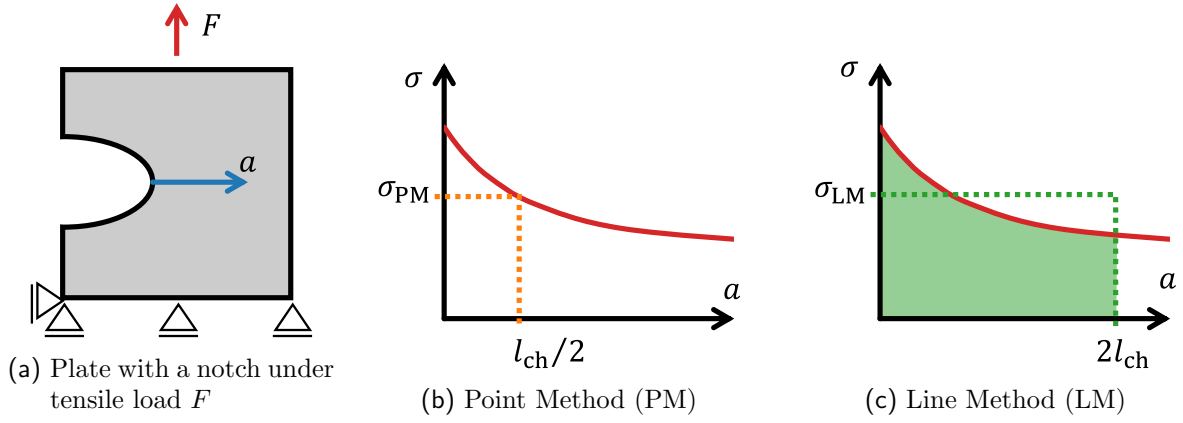


Figure 3.13: (a) The stress  $\sigma$  is extracted along a potential crack path with coordinate  $a$ , depicted by a blue arrow. (b) The Point Method (PM) uses  $\sigma$  at a certain distance  $a = l_{ch}/2$ . (c) The Line Method (LM) averages the stress  $\sigma$  from  $a = 0$  to  $a = 2 \cdot l_{ch}$

the distance for LM. The distances are chosen such that the stresses  $\sigma_{PM}$  and  $\sigma_{LM}$  are identical to the tensile strength  $\sigma_c$  if a critical uniaxial load is applied to a cracked infinite plate. Madrazo et al. [67] provides an analytical proof for this choice of distances. The LM and PM show a reasonable agreement with experimental results with a deviation below 20% [67].

In Paper D [34], the authors use a failure criterion to find critical positions on a hole surface. Next, the PM approach is used to predict crack initiation at the critical positions.

However, in some configurations LM and PM are unphysical. Cornetti et al. [31] demonstrated this for the example of a bending beam. For a bending beam with a height  $h = 2 \cdot l_{ch}$ , the average stress  $\sigma_{LM}$  of LM is always zero, because the compressive and tensile stresses cancel each other out. A similar thought experiment can be done for PM and a bending beam with a height  $h = l_{ch}$ . According to PM, the stress is evaluated at  $l_{ch}/2$  in the neutral axis, where the stress is zero. Consequently, PM and LM will never predict fracture in these cases, even for very high loads.

### 3.3.2 Energy Criterion

Instead of considering stresses, one can also analyze the strain energy released during crack initiation or crack propagation. According to the energy criterion, creating a new crack area  $A_c = b \cdot a_c$  requires a critical energy  $G_c \cdot A_c$ , where  $G_c$  is the critical energy release rate. Since energy cannot be generated, the energy stored in the new crack must be transferred from other energy components, like the kinetic energy  $\Pi_{kin}$  or the strain energy  $\Pi$ . Considering this, Leguillon [63] proposed the energy balance

$$\Delta\Pi + \Delta\Pi_{kin} + G_c \cdot A_c = 0 \quad (3.18)$$

for a newly created crack area  $A_c$ . In the static case, the change in the kinetic energy  $\Delta\Pi_{kin} \approx 0$  is neglected. Furthermore, it is common to rewrite Equation 3.18 and thereby

define the incremental energy release rate

$$G_{\text{inc}} := -\frac{1}{b} \frac{\Delta\Pi}{a_c} \geq G_c. \quad (3.19)$$

A crack of length  $a_c$  will initiate if the incremental energy release rate  $G_{\text{inc}}$  reaches the critical energy release rate  $G_c$ . However, an assumption for the crack length  $a_c$  is necessary.

### 3.3.3 Coupled Criterion

Leguillon [63, 68, 69] presented a Coupled Criterion (CC) that computes the crack length  $a_c$  automatically by coupling the stress and energy criterion. Let us assume, the incremental energy release rate  $G_{\text{inc}}(a, F)$  and the max. principal stress  $\sigma_I(a, F)$  are computed along a trial crack path  $a$  for an applied load  $F$ . The critical crack length  $a_c$  and the critical load  $F_c$  are two unknown parameters that can be found using two equations

$$\begin{cases} G_{\text{inc}}(a_c, F_c) \geq G_c \\ \sigma_I(a_c, F_c) \geq \sigma_c \end{cases} \quad (3.20)$$

from the energy and the stress criterion. The latter evaluates the stress in a critical distance  $a_c$ . Contrary to the PM approach within the TCD framework, the critical distance is not constant. Some authors [31, 70–72] prefer using the average stress in the style of the LM criterion. This leads to slightly different equations:

$$\begin{cases} G_{\text{inc}}(a_c, F_c) \geq G_c \\ \int_0^{a_c} \sigma_I(a, F_c) da \geq a_c \cdot \sigma_c \end{cases} \quad (3.21)$$

If a linear elastic material is used in conjunction with the small strain framework and linearized geometry, the incremental energy release rate scales quadratically with the applied load  $G_{\text{inc}} \propto F^2$  and the max. principal stress is directly proportional to the load  $\sigma_I \propto F$ . Consequently,  $G_{\text{inc}}$  and  $\sigma_I$  can be evaluated for an arbitrary load and then scaled to the critical load.

Figure 3.14 illustrates how to compute the critical load  $F_c = \lambda_c F$  for a critical load scale factor  $\lambda_c$  and an applied load  $F$ . A crack initiates at a hole once the energy and stress criterion are fulfilled simultaneously. The energy criterion (Fig. 3.14a) is fulfilled starting at a distance  $a_{\text{high}}$ . The stress criterion (Fig. 3.14b) is fulfilled up to a distance  $a_{\text{low}}$ . The load is then scaled by a load factor  $\lambda$ , until both criteria are fulfilled at a critical distance  $a_c$ , where  $a_{\text{high}} = a_{\text{low}}$ . From this intersection, depicted in Fig. 3.14c, the critical load for crack initiation is computed.

As mentioned before, the TCD approach is not physical for some bending beams. The CC approach, however, provides reasonable results for bending beams [31, 73, 74], sharp and blunted V-notches [70, 72, 75–77], circular holes [78, 79], and layered ceramics [80,

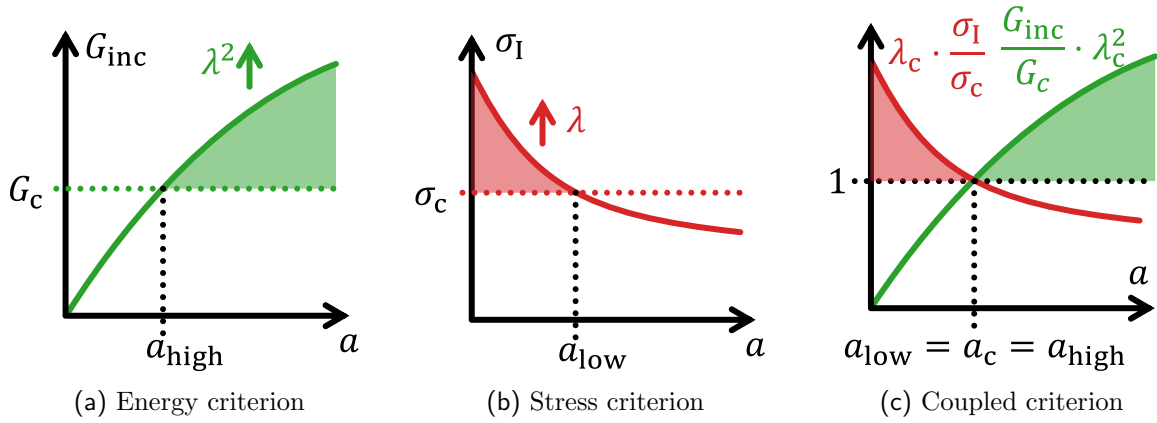


Figure 3.14: (a) According to the energy criterion, a crack may initiate with a length of at least  $a_{\text{high}}$  after  $G_{\text{inc}}$  exceeds  $G_c$ . (b) According to the stress criterion, a crack may initiate with a length up to  $a_{\text{low}}$  until  $\sigma_I$  drops below  $\sigma_c$ . (c) According to the Coupled Criterion (CC), a crack initiates once the energy and stress criterion are fulfilled simultaneously at a critical length  $a_c$ . This can be achieved by scaling the applied load by a factor  $\lambda$ , which increases  $G_{\text{inc}}$  by a factor of  $\lambda^2$  and  $\sigma_I$  by of  $\lambda$ .

81]. Figure 3.15 shows a selection of specimens with V-notches and circular holes under tensile loads. The incremental energy release rate,  $G_{\text{inc}}$ , and the max. principal stress,  $\sigma_I$ , are plotted as green and red curves, respectively.

A specimen with straight free edges can be considered as an open  $90^\circ$  V-notch. Figure 3.15a shows that such a specimen has a constant stress  $\sigma_I$  (red curve) over  $a$  and an incremental energy release rate  $G_{\text{inc}}$  (green curve) that increases linearly near the crack tip. If the specimen is wide enough,  $G_{\text{inc}}$  is high enough at some point and the energy criterion is fulfilled at lower loads than the stress criterion. Then only the stress criterion needs to be checked. However, for very thin specimens, the energy criterion is not yet fulfilled when the stress criterion is already fulfilled. Then the energy criterion needs to be checked. Parvizi et al. [82] observed this switch from the stress criterion to the energy criterion in cross-ply laminates with varying ply thicknesses.

The next model (Fig. 3.15b) is a quarter model of a specimen that contains a circular hole. Assuming an isotropic material, Inglis [37] stated that the stress at the hole,  $\sigma_{\text{max}}$ , is three times higher than the far field stress, due to the notch effect. The stress criterion is already fulfilled for low loads directly at the hole surface, but the energy criterion is only fulfilled further inside the specimen, when the stress has already decreased. Compared to a failure criterion like Rankine, which predicts fracture as soon as  $\sigma_{\text{max}}$  reaches  $\sigma_c$ , CC allows higher loads, where parts of the specimen are already above  $\sigma_c$ .

Figure 3.15c depicts a quarter model of a specimen with a  $30^\circ$  V-notch. There is a stress singularity at the notch tip and  $\sigma_I$  becomes infinite such that the stress criterion is fulfilled immediately [63]. The incremental energy release rate  $G_{\text{inc}}$  is zero at the tip and then rises rapidly until it reaches a plateau. Like for the previous specimens, CC is not fulfilled directly at the notch tip. Only at a finite distance from the notch tip,  $G_{\text{inc}}$  is high enough to meet the energy criterion.

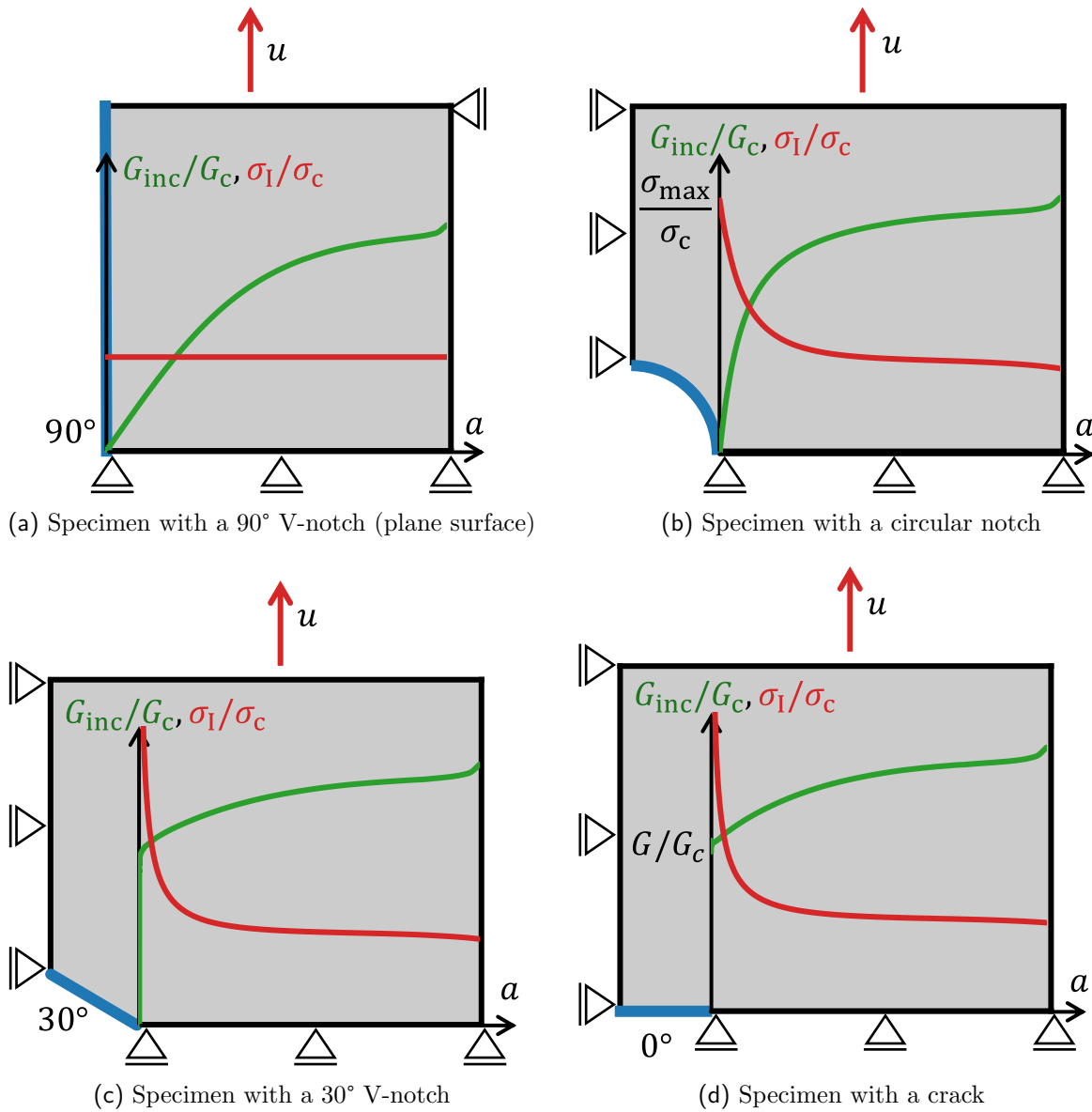


Figure 3.15: Quarter models of 2D specimens with various notches under uniaxial tension. The blue lines highlight the notches where cracks may initiate. The incremental energy release rate  $G_{\text{inc}}$  (green) and max. principal stress  $\sigma_I$  (red) are plotted along the path  $a$ .

For large cracks, the CC approach converges to the LEFM approach. This is shown in Fig. 3.15d. The specimen contains a crack, which is an extreme case of a V-notch with an opening angle of  $0^\circ$ . The stress is infinite at the crack tip and the stress criterion is fulfilled near the tip. Contrary to all other notches,  $G_{\text{inc}}$  is not zero at the crack tip. Instead, the incremental energy release rate  $G_{\text{inc}}$  at the crack tip becomes the differential energy release rate  $G$ , which is commonly used in LEFM [63]. Since the stress criterion is fulfilled at the tip, only the energy criterion needs to be checked at the crack tip. This approach corresponds to Griffith's  $G$ -concept.

### 3.3.4 Computation of the incremental energy release rate

The CC approach requires the incremental energy release rate  $G_{\text{inc}}$ . This section describes three approaches for the computation of  $G_{\text{inc}}$ . Like for the computation of the differential energy release rate, a M approach can be used. However, this can be too slow for realistic applications. For arbitrary smooth holes, the author developed the Scaling Law based Meta Model with Auto-Controlled boundary conditions (SLMM+AC) that is described in Paper C [83]. Furthermore, Leguillon [63] presented the Matched Asymptotics (MA) approach, which is suitable for V-notches.

#### 3.3.4.1 Full FEM approach for crack initiation

Full FEM is a common approach to compute  $G_{\text{inc}}$  numerically with a series of FEM simulations [33, 74, 83]. Like the Full FEM approach for the differential energy release rate  $G$ , a trial crack path  $a$  is introduced and the change in the strain energy is evaluated. However,  $G_{\text{inc}}(a)$  needs to be computed at more positions along the path  $a$ .

Figure 3.16 depicts this process. Starting from an uncracked specimen (blue) with thickness  $b$  and a strain energy  $\Pi_0$ , the trial crack path  $a$  is introduced normal to the surface. Next, the crack is opened in several steps (orange specimens). It is crucial to avoid remeshing, as this would lead to numerical errors. In each step, the strain energy  $\Pi_i$  and the crack length  $a_i$  are used to compute the incremental energy release rate  $G_{\text{inc}}(a_i)$  with Equation 3.19. The computed values are interpolated linearly. Furthermore, the incremental energy release rate for notches (contrary to cracks) starts at zero.

The Full FEM approach requires several FEM simulations to analyze a single position on a surface. If more positions or various load cases should be considered, the number of simulations increases significantly.

#### 3.3.4.2 Scaling Law based Meta Model

To avoid a huge number of simulations, the author presented in his master thesis [84] a Scaling Law based Meta Model (SLMM) for 3D holes. In Paper A [83], this approach is extended to the SLMM+AC approach which is more accurate and suitable for 2D holes.

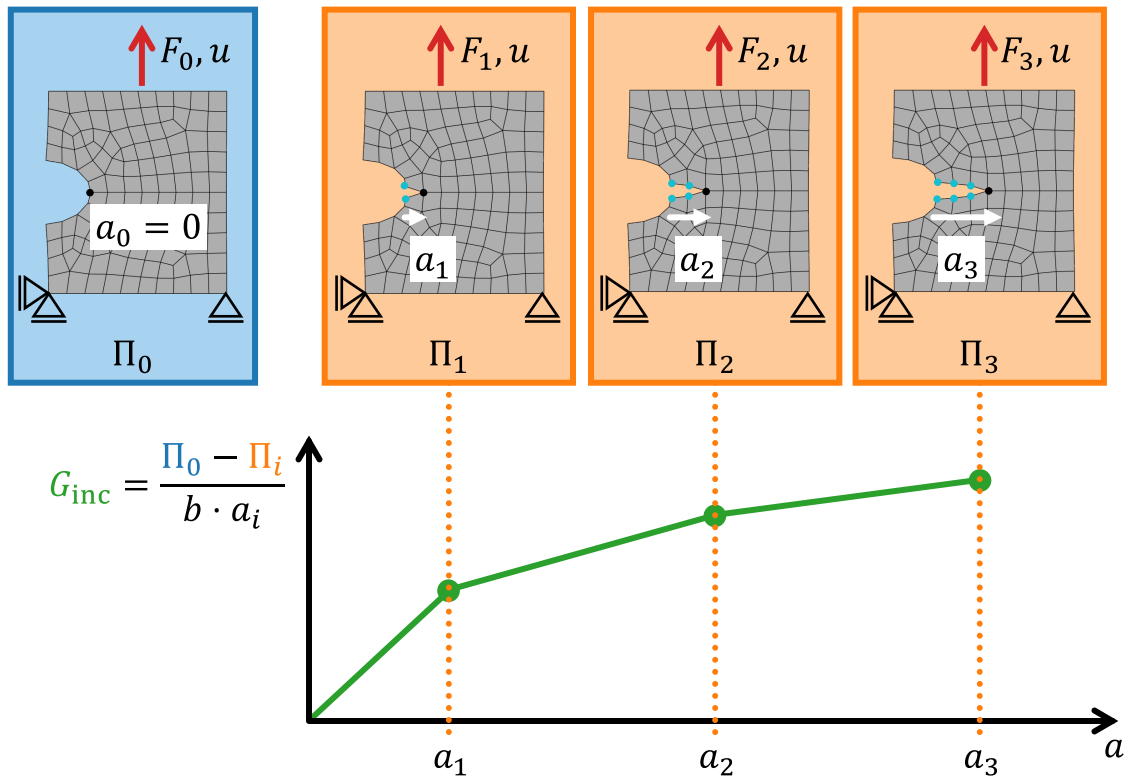


Figure 3.16: Computation of the incremental energy release rate  $G_{inc}$ . The blue specimen shows the initial state without a crack and a strain energy  $\Pi_0$ . The orange specimens depict the specimen with a growing crack for a fixed displacement  $u$ . The corresponding strain energy  $\Pi_i$  and the crack increment  $a_i$  are used to compute  $G_{inc}$ .

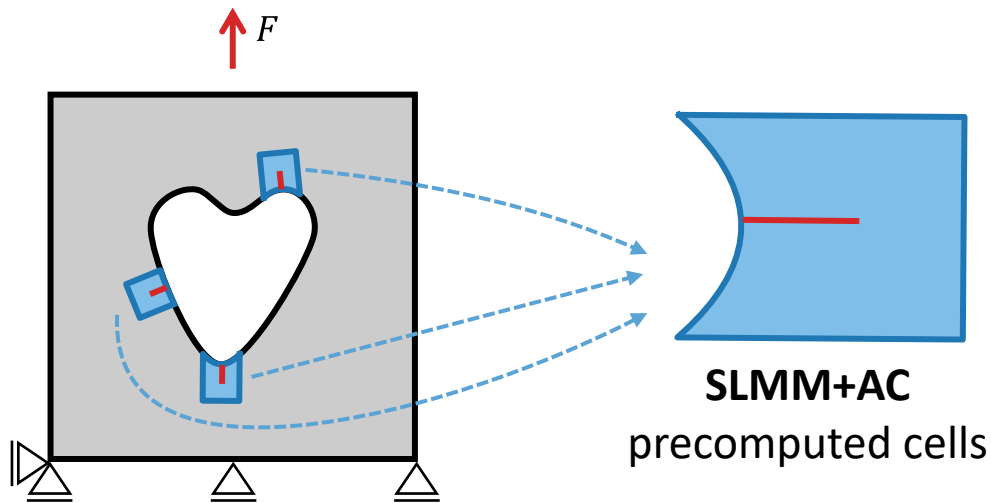


Figure 3.17: The Scaling Law based Meta Model with Auto-Controlled boundary conditions (SLMM+AC) uses precomputed cells to approximate  $G_{inc}(a)$  and  $\sigma_I(a)$ . The precomputed cells are scaled to match any smooth position at a hole surface.

The SLMM approach is a two-scale approach. The larger scale corresponds to the specimen level with smooth holes, as depicted on the left in Fig. 3.17. The specimen is simulated using one FEM simulation to obtain the displacement field. The smaller scale contains precomputed cells for  $\sigma_I(a)$  and  $G_{inc}(a)$ , as depicted on the right in Fig. 3.17.



These precomputed cells can be aligned to an arbitrary position on the hole and the precomputed results, stored in the meta model, can be scaled to obtain an approximation of  $G_{\text{inc}}(a)$  and  $\sigma_{\text{I}}(a)$  at this position. For this transfer of the results from the precomputed cell to the specimen, methods like *scaling laws*, *load superposition*, *meta modeling*, and *auto-controlled boundary conditions* are used.

**Meta model** The precomputed cells have a unconstrained curved edge on the left side at which a trial crack (red) initiates. The top, right, and bottom sides have prescribed displacements for a given load case. A set of load cases like tension, shear, and bending is defined. The precomputed cell is parametrized by the curvature radius of the curved side, the load case, and the Poisson's ratio. Next,  $G_{\text{inc}}(a)$  and  $\sigma_{\text{I}}(a)$  are computed for all combinations of these parameters. The results are stored in a meta model, such that the results can be accessed at a later point based on these three parameters. Parameter values not stored in the meta model are interpolated piecewise linearly.

**Scaling laws** The precomputed cells have two independent dimensions: A length dimension (mm) and a stress dimension (MPa). According to dimensional analysis [85] these two independent dimensions can be used to freely chose two quantities with the corresponding dimension. Those two quantities are the curvature radius of the free side for the length dimension (mm) and the Young's modulus for the stress dimension (MPa). This allows to fit the precomputed cell to a hole with an arbitrary curvature radius and further SLMM can predict results for an arbitrary Young's modulus. The results  $G_{\text{inc}}(a)$  and  $\sigma_{\text{I}}(a)$  are then scaled in common with the Young's modulus and the curvature radius.

**Load superposition** Results are stored in the meta model for the independent load cases. However, a mixture of load cases is present inside the specimen. According to the load superposition principle, the load mixture is represented by a weighted sum of the independent load cases and thus  $G_{\text{inc}}(a)$  and  $\sigma_{\text{I}}(a)$  can be computed by a weighted sum of the stored results.

**Auto-controlled boundary conditions** A drawback of the SLMM approach is that it behaves like a sub-model with displacement-controlled boundary conditions. If a crack initiates, the stiffness of the cell decreases, but a global stress redistribution over the whole specimen is not possible, due to the prescribed displacements. Consequently, SLMM underestimates  $G_{\text{inc}}(a)$  and only works for cracks that are small compared to the precomputed cell. An alternative to displacement-controlled ( $u$ -controlled) boundary conditions are force-controlled ( $F$ -controlled) boundary conditions. However, a  $F$ -controlled approach would lead to an overestimated  $G_{\text{inc}}(a)$ .

Auto-controlled boundary conditions compute a weighting factor between the  $u$ -controlled and  $F$ -controlled results. For this, two precomputed cells (a smaller and a larger one) are

placed on the same position at a surface, as depicted in Fig. 3.18. For both precomputed cells,  $G_{\text{inc}}(a)$  is computed  $u$ -controlled and  $F$ -controlled. Next,  $G_{\text{inc}}(a)$  is computed as a weighted sum of the  $u$ -controlled and  $F$ -controlled results. The weighting factor is found, such that  $G_{\text{inc}}(a)$  of both cells overlap. This leads to a more accurate approximation of  $G_{\text{inc}}(a)$ .

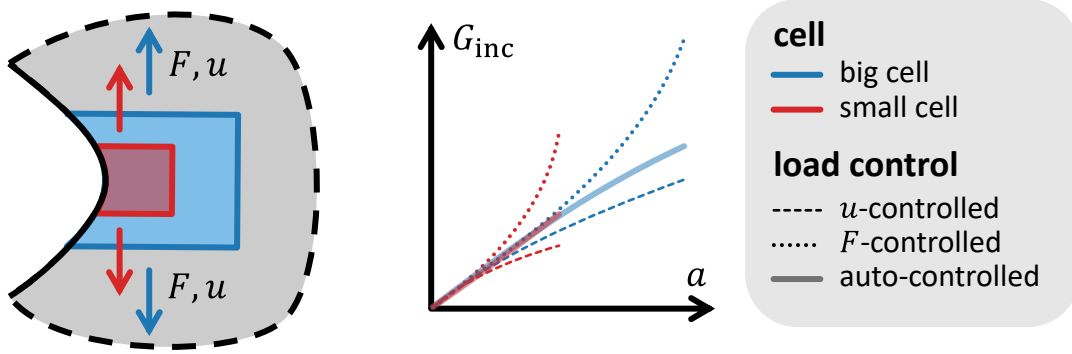


Figure 3.18: The left images shows a notch. A big cell (blue) and a small cell (red) are positioned at the notch surface. Both cells predict  $G_{\text{inc}}(a)$  using a displacement  $u$ - and force  $F$ -controlled approach. The auto-controlled approach weights the  $u$ - and  $F$ -controlled curves such that they match for the small and big cells.

### 3.3.4.3 Matched Asymptotics

The SLMM+AC approach works for smooth holes. For sharp V-notches, Leguillon [63] proposed the Matched Asymptotics (MA) approach. Later, it was extended to blunted V-notches [64, 74, 76]. The basic idea of MA is to use a generalized stress intensity factor  $k_{\text{MA}}$ , a singularity exponent  $\lambda_{\text{MA}}$ , and a geometrical coefficient  $B$  to compute the incremental energy release rate [74]

$$G_{\text{inc}}(a) = k_{\text{MA}}^2 \cdot a^{2 \cdot \lambda_{\text{MA}} - 1} \cdot B \quad (3.22)$$

and the stress

$$\sigma(a) = k_{\text{MA}} \cdot a^{\lambda_{\text{MA}} - 1}. \quad (3.23)$$

The computation of  $k_{\text{MA}}$ ,  $\lambda_{\text{MA}}$ , and  $B$  is not straight forward, but Leguillon [63] provides a table for  $\lambda_{\text{MA}}$  in dependence of the opening angle of the V-notch. The singularity exponent is  $\lambda_{\text{MA}} = 0.5$  for a crack and  $\lambda_{\text{MA}} = 1$  for a fully opened V-notch. An interesting observation is that  $G_{\text{inc}}$  always starts at zero, except for a crack with  $\lambda_{\text{MA}} = 0.5$ . For a crack,  $G_{\text{inc}}(0)$  becomes the differential energy release rate  $G$ , as illustrated by Fig. 3.15d. Contrary, the stress is always infinite at  $a = 0$ , except for  $\lambda_{\text{MA}} = 1$ , which corresponds to the fully opened V-notch, as depicted in Fig. 3.15a.

## 4 Toughness optimization

The aim of this thesis is to find geometries that maximize the tensile toughness

$$U_T = \int_0^{u_{\max}} F(u) du \quad (4.1)$$

that is the work needed to fully break a 2D plate loaded by uniaxial tension, as depicted in Fig. 4.1. There are several ways to improve the tensile toughness of a structure: e.g., by using lattice structures [13, 15], introducing inhomogeneities at different scales like in natural material as bones or nacre [3]. In this work, the author purely focuses on introducing holes on a macro-scale above one millimeter. Several research groups demonstrated that the position and shape of holes significantly influences the tensile toughness [7, 9, 10]. Yadav et al. [9] found that slits (elongated holes) lead to one of the toughest structures if the crack propagates perpendicular to the slits. However, for cracks propagating parallel to the slits, these slits themselves may even act as cracks and thus drastically reduce the strength and the tensile toughness.

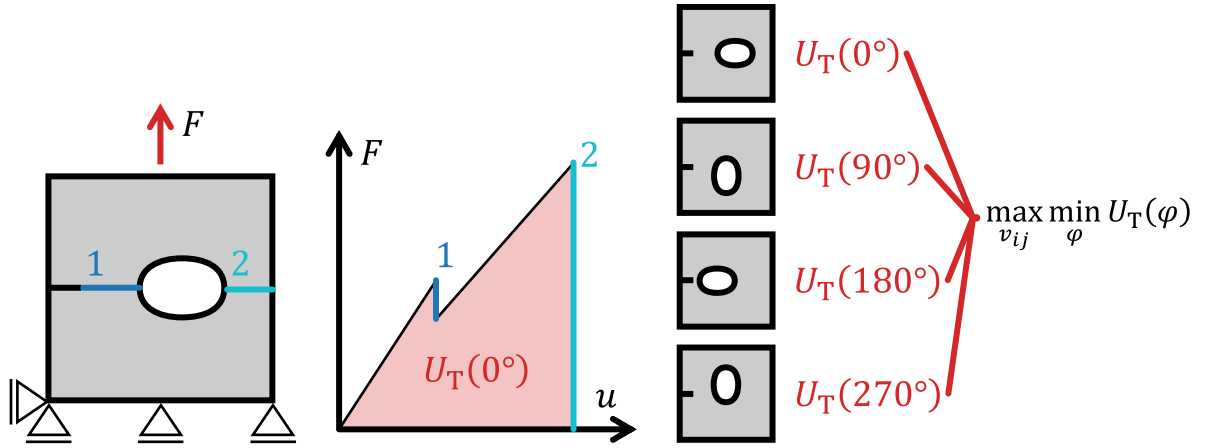


Figure 4.1: Maximizing the minimum tensile toughness of a plate under uniaxial tension by varying the hole shapes. The minimum tensile toughness is computed from the specimens with the hole shapes rotated in  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$ , and  $270^\circ$ .

That is why the approach, presented in Paper D [34], aims to maximize the tensile toughness for varied crack propagation directions. Figure 4.1 illustrates this objective. A uniaxial tensile load is applied to a 2D plate that contains an initial crack in the center of the left edge. The hole shape should be optimized. In the shown plate, the crack first propagates from the initial crack into the hole. After this event, marked as **1**, a new crack, marked as **2**, initiates at the hole. The crack propagates to the right side and fully

breaks the specimen. During this fracture process, the force-displacement  $F(u)$  curve is determined and the tensile toughness  $U_T$  is computed. The procedure is repeated for holes that are rotated by  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$ , and  $270^\circ$ . Each time the tensile toughness  $U_T(\varphi)$  is computed. Next, the lowest of these tensile toughness  $\min_{\varphi}(U_T(\varphi))$  is considered. This lowest tensile toughness corresponds to a crack approaching from the worst side and this lowest tensile toughness should be maximized.

One can make a few considerations to manually optimize the hole shape. In general, the area under the  $F(u)$  curve should be maximized. This can be done by increasing the force or the displacement. Typical patterns to increase the toughness are circular holes aligned in a row or a grid [9, 10], as depicted in Fig. 4.2. The holes stop cracks and new cracks have to initiate at higher forces. However, holes reduce the overall stiffness and subsequently the forces. Consequently, there is a tradeoff between more holes which stop cracks more often and a higher overall stiffness. Liu [10] performed a parameter study and highlighted some circular hole alignments that have a higher toughness than a solid plate.

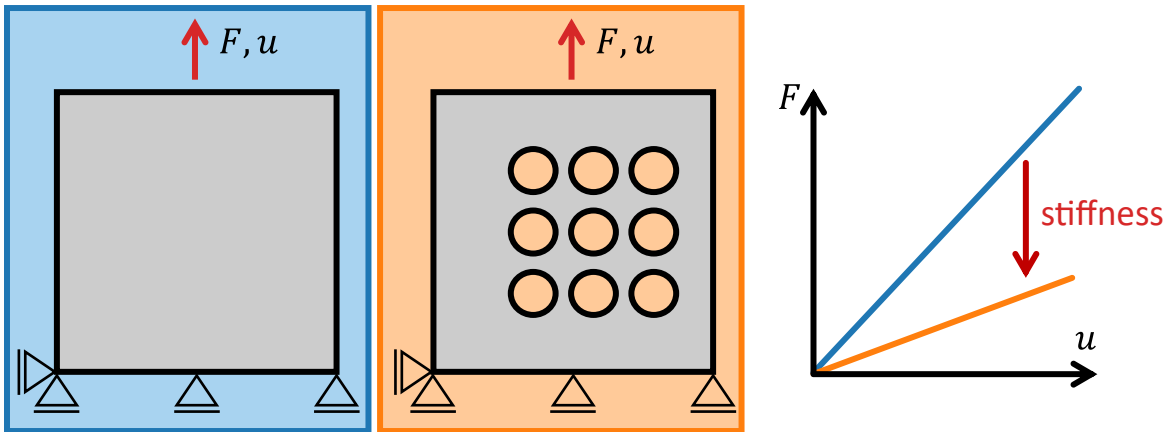


Figure 4.2: Removing material by adding new holes reduces the stiffness.

Furthermore, the hole shape plays a crucial role, as illustrated by Fig. 4.3. Yadav et al. [9] investigated various hole shapes like slits and circular holes. They found that slits in loading direction (blue specimen in Fig. 4.3) stop cracks and require the highest work to break the specimen compared to the investigated geometries. Contrary, a slit perpendicular to the loading direction, as depicted in orange in Fig. 4.3, reduces the critical load due to the notch effect [71].

As the slit becomes thinner, its curvature radius becomes smaller. Damani et al. [86] showed in this case that the critical load also becomes smaller until a critical curvature radius at which the slit can be seen as a crack [2, 37]. Figure 4.4 illustrates the much lower critical load for such a cracked specimen (orange) compared to a solid specimen (blue) with the same cross section.

Slits aligned parallel to the loading direction increase the tensile toughness, whereas cracks and slits perpendicular to the loading direction reduce it. Consequently, slits can be

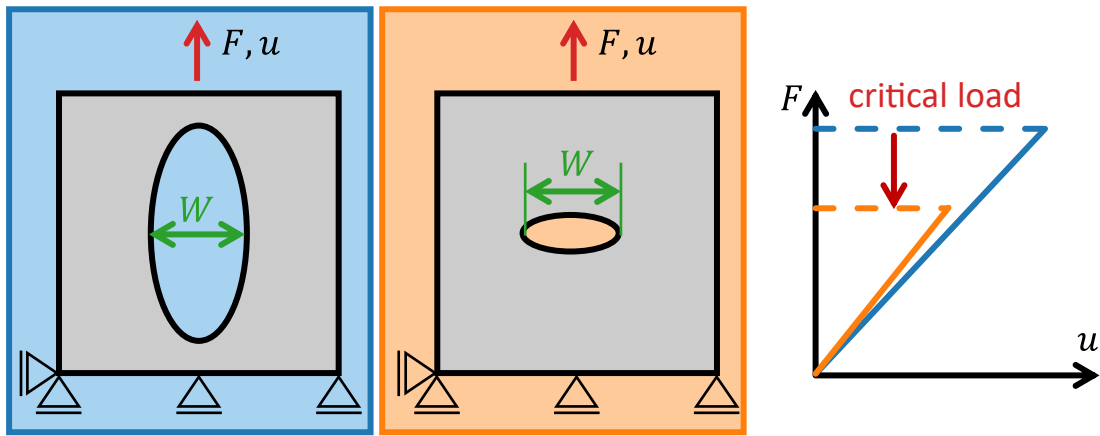


Figure 4.3: An elongated hole (slit) reduces the critical load at which a crack initiates due to the notch effect.

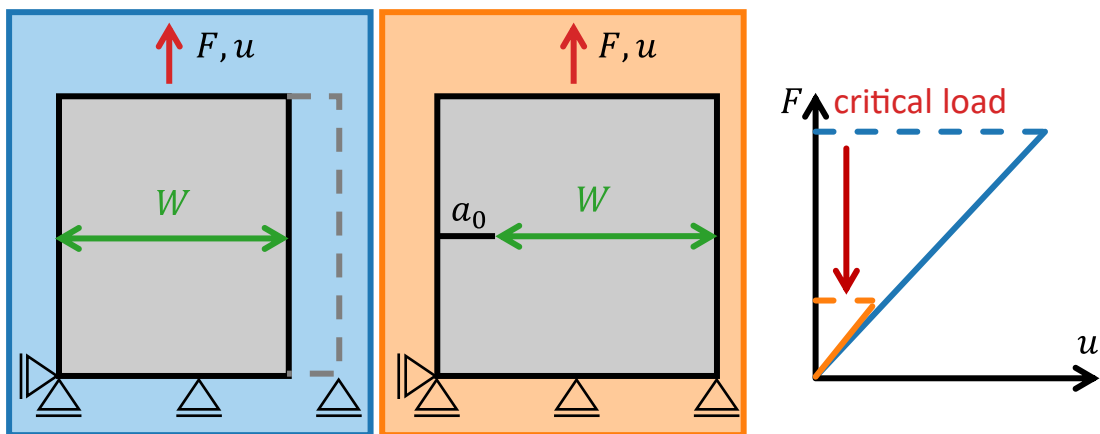


Figure 4.4: A crack greatly reduces the critical load.

only optimized for one crack propagation direction and lead to a worse tensile toughness in the  $90^\circ$  rotated case. An alternative to slits that avoid this issue are stress relief notches or holes [87]. Figure 4.5 qualitatively compares the  $F(u)$  curve of a specimen with a single circular hole (blue) with a specimen that contains two additional stress relief holes above and below the hole in the center (orange). First, the crack propagates, starting from the initial crack. This occurs at the first drop in the  $F(u)$  curves. Due to the additional holes, the stiffness of the orange specimen will be slightly smaller. Next, a new crack initiates at the center hole and here, the critical force of the orange specimen is greater, because the two stress relief holes deflect the stress from the centered hole. Consequently, the orange specimen has a higher tensile toughness.

Stress relief holes can also lead to stable crack propagation or even crack arrest. In most scenarios, a crack will propagate in an unstable manner in tensile tests. That means that the crack continues to propagate without further load increase until it hits a surface. The blue specimen in Fig. 4.6 shows such an unstable crack propagation that appears in the  $F(u)$  curve as a drop in the force at a certain displacement. However, during the work on Paper D [34], the authors found some hole geometries where a crack propagates

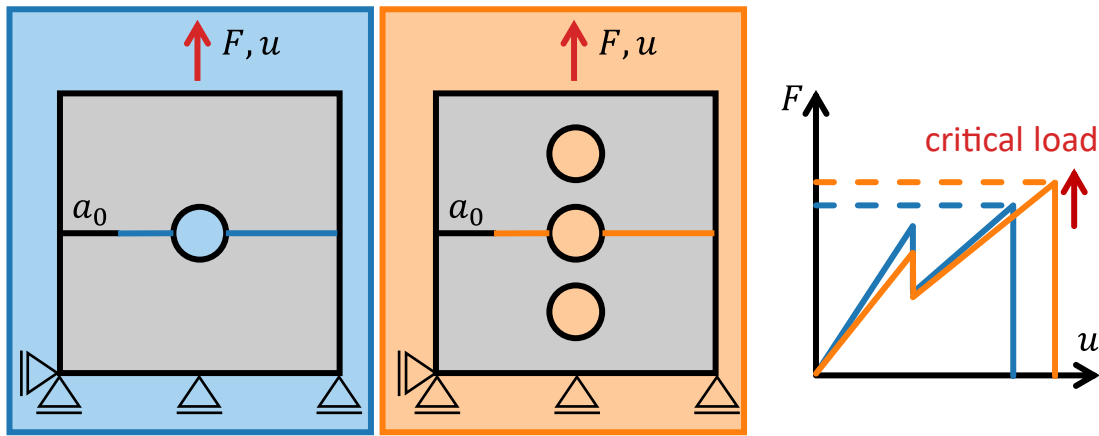


Figure 4.5: Additional holes placed above and below the center hole relieve the stress at the center hole and lead to a higher critical load for the crack reinitiation (second peak). The first peak corresponds to the crack propagation of the initial crack.

in a stable manner or even arrests. The orange specimen in Fig. 4.6 shows such a case. The specimen contains two stress relief holes above and below the crack path. The crack starts propagating from the initial crack. This causes the first drop in the orange  $F(u)$  curve. Then, the crack propagates between the two holes. Since the two holes shield the stress from the crack tip, a higher critical load is required, which leads to a stable crack propagation. At some point, the critical load for the crack propagation becomes even higher than the critical load for the initiation of a new crack. This point (2) depends on the material parameters and the exact geometry. The original crack then stops (arrests) and the new crack propagates in an unstable manner into the right surface. After this, the original crack finally propagates further into the top hole.

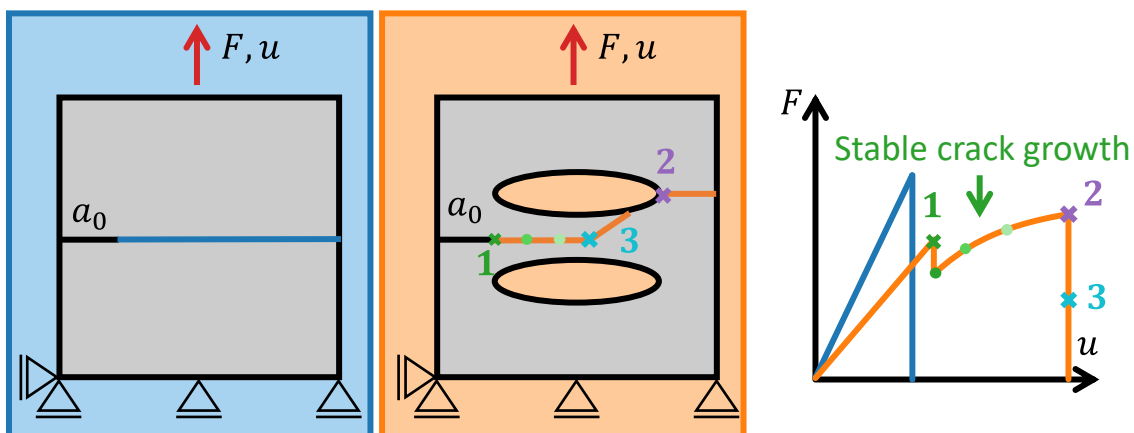


Figure 4.6: Unstable crack growth (blue) and stable crack growth (orange): For the orange specimen, the crack starts growing (1), which causes the first drop in the force. Then, the crack stops (2), because the holes above and below shield the crack tip and an initiation at the purple cross is favored. After the newly initiated crack grows into the right edge, the first crack starts again (3) and grows into the top hole.

In addition to these considerations, an optimization algorithm was used to find tough structures. For the optimization process, two crucial issues were addressed. The first issue

is the highly non-linear objective to maximize the tensile toughness. This objective has multiple local maxima and the evaluation of one hole design takes several minutes. Consequently, an efficient optimization algorithm is required which can escape local maxima. The second issue is the representation of arbitrary holes. For optimization, it needs to be possible to create new holes with smooth contours or modify and delete existing holes.

## 4.1 Optimization algorithm

In this section, the "Truncated Breadth First Search with Limited depth" (TBFSL) optimization algorithm developed by the author and published in Paper C [88] is described. Common topology and structural optimization methods work well for objectives like minimizing the compliance [89], but they do not perform well for objectives with many local optima. To optimize the tensile toughness, a global optimization algorithm is required that can escape local maxima and requires as few evaluations as possible.

Variable Neighborhood Search (VNS) [90–92] algorithms are such global optimization methods. The idea of VNS is to explore the neighborhood of the currently best design. If no improvement was achieved in this neighborhood, the neighborhood expands until a better design is found or the termination criterion is met.

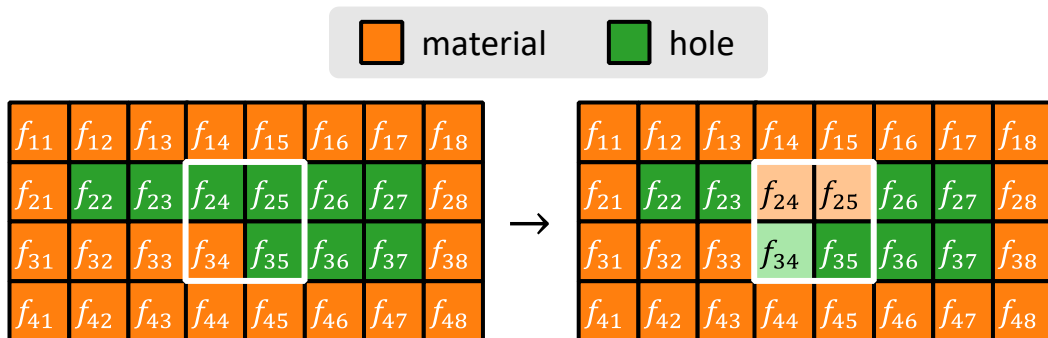


Figure 4.7: TBFSL iteration: The optimization variables  $f_{ij}$  are aligned in a grid and are assigned either to a material or a hole. In this iteration, up to three variables vary in a rectangular region (white framed) with an edge length of 2.

The TBFSL algorithm follows the idea of VNS and is suitable for spatial 2D problems. It aligns binary variables into a 2D grid, as depicted in Fig. 4.7. The variables represent either a material or a hole. Starting from an initial design, all designs in a certain neighborhood are evaluated until an improvement is found. The neighborhood is defined by two parameters: the size of the region to modify  $\Delta r_{\max}$  and the number of differing variables  $\Delta d_{\max}$ . The latter describes how many variables can be changed in one iteration. The region size  $\Delta r_{\max}$  corresponds to the perimeter of a rectangle. All variables changed in one iteration have to lie inside a rectangle of this size  $\Delta r_{\max}$ . In the case of Fig. 4.7,  $\Delta d = 3$  variables change inside a rectangle of size  $\Delta r = 2$ . Note, that TBFSL first starts iterating designs closest to the current design with  $\Delta d = 1$ . If no improvement is

found, the neighborhood expands to  $\Delta d = 2$  and so on until it exceeds  $\Delta d_{\max}$ . Then, the optimization terminates.

In Paper C [88], the authors showed that TBFSL performs better than alternative common global optimization algorithms like Genetic Algorithms [93–96] and Simulated Annealing [97]. The algorithms TBFSL, GA, and SA were assessed with an example problem: the endpoint displacement of a bending beam loaded by its own weight under gravity was minimized. Figure 4.8 depicts the bending beam that is fully constrained on the left side. The endpoint displacement  $u$  is measured at the top right corner. The dark gray squares are fixed to be material to guarantee a connection to the top right corner.

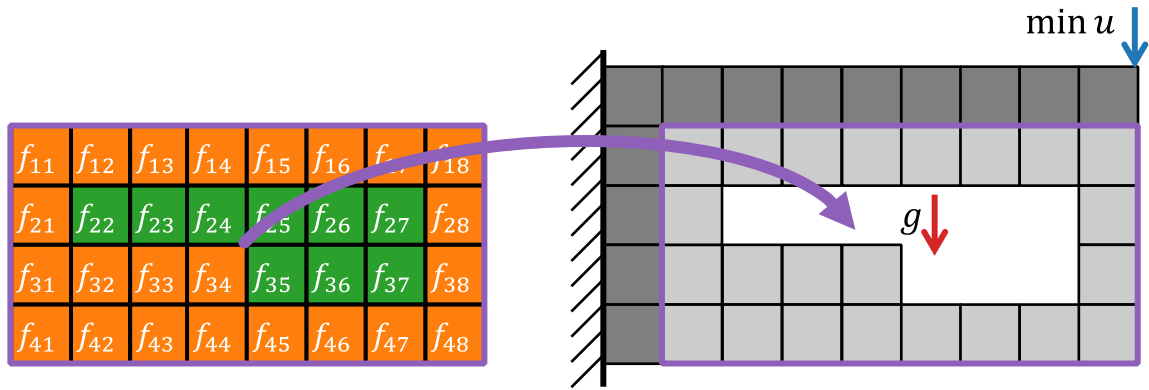


Figure 4.8: Minimizing the endpoint displacement  $u$  of a cantilever beam loaded by its own weight under gravity  $g$ . The design space is highlighted by a violet frame.

The authors found that TBFSL is more than ten times more efficient than GA and SA, in the sense that it needs more than ten times fewer iterations for the same result. For this reason, TBFSL was chosen for the tensile toughness optimization problem.

## 4.2 Representation of hole shapes

In the previous section, geometries represented by square pixels were optimized. Such designs are not suitable to maximize the tensile toughness, because each sharp corner of a square pixel acts as notch, which decreases the toughness. Consequently, a method to generate smooth holes is required.

Figure 4.9 illustrates the smoothing approach used in Paper D [34]. The optimization algorithm alters variables  $f_{ij}$  aligned in a 2D grid. These variables are assigned to so-called support points which are shown as spheres in Fig. 4.9a. These support points are interpolated using Radial Basis Functions (RBF) [98, 99]. The RBF approach interpolates a value at a certain position "P" by a weighted sum of all support point values. The weighting factors depend on the distance of the position "P" to the support points. Support points closer to "P" have more influence than support points far away from "P".

In the next step, a new plane, depicted in grey in Fig. 4.9b, is defined at a level between the orange (material) and green (hole) values. The hole shapes are then generated by



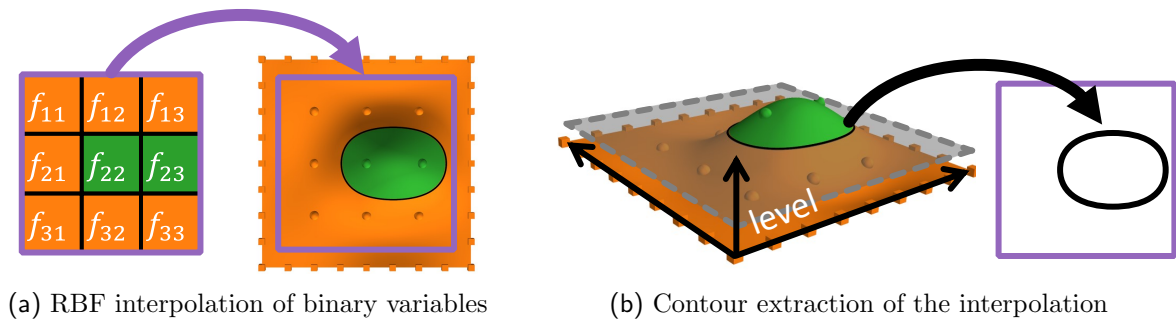


Figure 4.9: Generating smooth geometries from the variable grid. (a) The variables are interpolated. (b) The interpolation is cut at a certain level and the contour (black line) is used as hole geometry.

intersecting the RBF interpolation with the plane.

A drawback of this approach is that the hole surface can never cross the position of the support points. To compensate this, the optimization is performed multiple times with support points aligned in various grids.

### 4.3 Optimization result

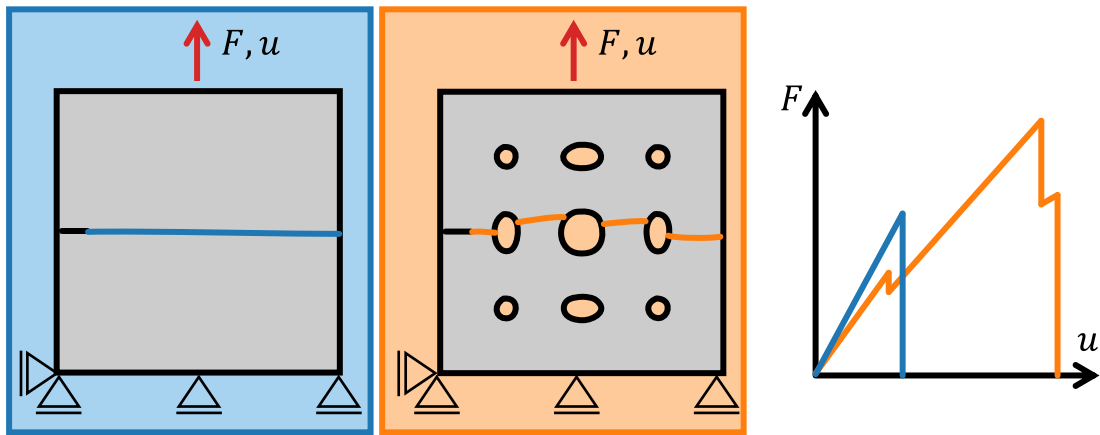


Figure 4.10: Start plate (blue) and plate (orange) optimized in Paper D [34] with the  $F(u)$  curve.

In Paper D [34], the TBFSL algorithm and the representation of holes shapes are used as described before to optimize the tensile toughness of a PMMA plate. All optimization runs started with a solid plate, as depicted blue in Fig. 4.10. The orange plate is the result of the optimization. It has several holes into which the crack propagates such that it has to reinitiate at a higher load. The critical load for crack reinitiation is further increased by stress relief holes that are placed above and below the initiation positions. The  $F(u)$  curve of the orange plate shows a much tougher material response compared to the blue curve. The tensile toughness of the orange plate is about 4.5 times higher than the tensile toughness of the blue plate.



## 5 Compilation of the thesis

The second part of this thesis contains the following appended papers. Paper A-Paper C are published in peer-reviewed journals under the creative commons license CC-BY 4.0. Paper D has been submitted to a journal and is under review. Table 5.1 lists the contributions of the author.

**Paper A:** M. Rettl, M. Pletz, C. Schuecker, Efficient prediction of crack initiation from arbitrary 2D notches, *Theoretical and Applied Fracture Mechanics* 119 (2022) 103376. 10.1016/j.tafmec.2022.103376.

**Paper B:** M. Rettl, S. Frankl, M. Pletz, M. Tauscher, C. Schuecker, ConForce: Computation of configurational forces for FEM results, *SoftwareX* 26 (2024) 101718. 10.1016/j.softx.2024.101718.

**Paper C:** M. Rettl, M. Pletz, C. Schuecker, Evaluation of combinatorial algorithms for optimizing highly nonlinear structural problems, *Materials & Design* 230 (2023) 111958. 10.1016/j.matdes.2023.111958.

**Paper D:** M. Rettl, M. Pletz, C. Schuecker, Optimizing the hole geometry of 2D plates for maximum tensile toughness, 10.2139/ssrn.4962118. *under review in Engineering Fracture Mechanics*.

Table 5.1: Contribution of the author to the papers.

Contributions	Paper A	Paper B	Paper C	Paper D
Conceptualization	80 %	50 %	80 %	60 %
Method development & implementation	100 %	60 %	100 %	60 %
Numerical study	100 %	80 %	100 %	70 %
Writing – Original Draft	80 %	80 %	80 %	60 %



---

## 6 Conclusions

Damage-tolerant materials are required for many engineering applications. This work presents a method to make specimens more damage-tolerant. This can be done by optimizing the tensile toughness, which is the work required to fully break a specimen. To compute the tensile toughness, the whole fracture process including crack initiation and propagation must be predicted. Crack initiation can be predicted by the Coupled Criterion that involves the incremental energy release rate. However, the incremental energy release rate is computationally expensive to compute, especially if the initiation position is not known in advance. In Paper A, the author developed a scaling law based meta model that is able to efficiently and accurately approximate the incremental energy release rate and subsequently predict crack initiation. Since the presented approach is very efficient, it can be applied to every possible initiation position and state the most critical one.

After a crack has initiated, the crack propagates in a certain direction once the critical load is reached. In this work, the direction of the maximum energy release rate is assumed to be the crack propagation direction. Like for the crack initiation, this requires the computation of an energy release rate. Contrary to the crack initiation, now the differential energy release rate is required which can be computed using FEM simulations. However, this needs to be done for every possible propagation direction to find the direction of the maximum energy release rate. Many FEM simulations would be required to do so. As a more efficient alternative approach, the J-Integral can be used to approximate the propagation direction. More precisely, Configurational Forces are used to compute the J-Integral, because Rice's original formulation of the J-Integral cannot be computed using FEM results. The implementation of the Configurational Forces is available as a Python package and as Abaqus Plugin, as presented in Paper B.

With the developed methods, it is possible to predict the fracture process and evaluate the tensile toughness. However, optimizing the tensile toughness is a highly non-linear problem with many local maxima. In Paper C, the authors developed the "Truncated Breadth First Search with Limited depth" (TBFSL) optimization method that is suitable for such a problem. The idea of TBFSL is to search for better solutions in the neighborhood of the current solution. If no better solution is found, TBFSL extends the search radius in the neighborhood. The TBFSL algorithm was assessed against common global optimization algorithms like Genetic Algorithms and Simulated Annealing. For this, a test problem was defined: the endpoint displacement of a bending beam loaded by its own weight should be minimized by adding and removing material in the design space. To

obtain a similar solution, TBFSL needs on average a lower number of FEM simulations compared to the Genetic Algorithm and Simulated Annealing.

Finally, Paper D describes the optimization of the tensile toughness of a 2D plate with arbitrary-shaped holes. The fracture process was predicted using the Configurational Forces implementation of Paper D. The TBFSL optimization algorithm maximized the tensile toughness and yielded designs with a tensile toughness that is more than 4.5 times higher compared to a solid design.

This work is a first step in understanding toughening mechanisms as they occur in natural materials. Based on observations in natural materials, the author demonstrated a method to optimize the toughness of engineering materials. The author showed that the toughness can be increased even for brittle base materials by adding holes. These holes can stop crack propagation, which is an important toughening mechanism. In contrast to the problem investigated in this work, many natural materials combine several materials with various properties. Such natural materials can be seen as guideline to further increase the toughness of engineering materials. For example, one could extend this work and add inclusions instead of holes to a base material. The inclusion could be a softer material. Furthermore, it is important to validate the predictions of the fracture process with real experiments.

---

## 7 References

- [1] R. Danzer, T. Lube, P. Supancic and R. Damani. ‘Fracture of Ceramics’. In: *Advanced Engineering Materials* 10.4 (Apr. 2008), pp. 275–298. DOI: 10.1002/adem.200700347.
- [2] T. L. Anderson. *Fracture Mechanics: Fundamentals and Applications*. CRC Press, 1991.
- [3] R. O. Ritchie. ‘The Conflicts between Strength and Toughness’. In: *Nature Materials* 10.11 (Nov. 2011), pp. 817–822. DOI: 10.1038/nmat3115.
- [4] Y. Chao, J. Ward and R. Sands. ‘Charpy Impact Energy, Fracture Toughness and Ductile–Brittle Transition Temperature of Dual-Phase 590 Steel’. In: *Materials & Design* 28.2 (Jan. 2007), pp. 551–557. DOI: 10.1016/j.matdes.2005.08.009.
- [5] P. C. Okonkwo et al. ‘A Focused Review of the Hydrogen Storage Tank Embrittlement Mechanism Process’. In: *International Journal of Hydrogen Energy* 48.35 (Apr. 2023), pp. 12935–12948. DOI: 10.1016/j.ijhydene.2022.12.252.
- [6] U. Hejman and C. Bjerken. ‘Dissolution Driven Crack Branching in Polycarbonate’. In: *Fatigue & Fracture of Engineering Materials & Structures* 34.4 (Apr. 2011), pp. 227–239. DOI: 10.1111/j.1460-2695.2010.01508.x.
- [7] D. Brescakovic and O. Kolednik. ‘Fracture Toughness Improvement Due to Crack Deflection and Crack Trapping by Elliptical Voids or Particles’. In: *International Journal of Solids and Structures* 285 (Dec. 2023), p. 112551. DOI: 10.1016/j.ijsolstr.2023.112551.
- [8] D. Brescakovic, M. Kegl and O. Kolednik. ‘Interaction of Crack and Hole: Effects on Crack Trajectory, Crack Driving Force and Fracture Toughness’. In: *International Journal of Fracture* 236.1 (July 2022), pp. 33–57. DOI: 10.1007/s10704-021-00611-1.
- [9] D. Yadav, T. More and B. N. Jaya. ‘Morse-Code Inspired Architectures for Tunable Damage Tolerance in Brittle Material Systems’. In: *Journal of Materials Research* 37.6 (Mar. 2022), pp. 1201–1215. DOI: 10.1557/s43578-022-00520-6.
- [10] Y. Liu, L. St-Pierre, N. Fleck, V. Deshpande and A. Srivastava. ‘High Fracture Toughness Micro-Architected Materials’. In: *Journal of the Mechanics and Physics of Solids* 143 (Oct. 2020), p. 104060. DOI: 10.1016/j.jmps.2020.104060.

- 
- [11] S. Yin, H. Chen, R. Yang, Q. He, D. Chen, L. Ye, Y.-W. Mai, J. Xu and R. O. Ritchie. ‘Tough Nature-Inspired Helicoidal Composites with Printing-Induced Voids’. In: *Cell Reports Physical Science* 1.7 (July 2020), p. 100109. DOI: 10.1016/j.xcrp.2020.100109.
- [12] S. Choukir and C. Singh. ‘Role of Topology in Dictating the Fracture Toughness of Mechanical Metamaterials’. In: *International Journal of Mechanical Sciences* 241 (Mar. 2023), p. 107945. DOI: 10.1016/j.ijmecsci.2022.107945.
- [13] N. A. Fleck and X. Qiu. ‘The Damage Tolerance of Elastic–Brittle, Two-Dimensional Isotropic Lattices’. In: *Journal of the Mechanics and Physics of Solids* 55.3 (Mar. 2007), pp. 562–588. DOI: 10.1016/j.jmps.2006.08.004.
- [14] K. M. Conway, C. Kunka, B. C. White, G. J. Pataky and B. L. Boyce. ‘Increasing Fracture Toughness via Architected Porosity’. In: *Materials & Design* 205 (July 2021), p. 109696. DOI: 10.1016/j.matdes.2021.109696.
- [15] O. Al-Ketan, R. Rowshan and R. K. Abu Al-Rub. ‘Topology-Mechanical Property Relationship of 3D Printed Strut, Skeletal, and Sheet Based Periodic Metallic Cellular Materials’. In: *Additive Manufacturing* 19 (Jan. 2018), pp. 167–183. DOI: 10.1016/j.addma.2017.12.006.
- [16] A. Dasari, Q.-X. Zhang, Z.-Z. Yu and Y.-W. Mai. ‘Toughening Polypropylene and Its Nanocomposites with Submicrometer Voids’. In: *Macromolecules* 43.13 (July 2010), pp. 5734–5739. DOI: 10.1021/ma100633y.
- [17] H. Kim, H. Keskkula and D. Paul. ‘Effect of Acrylonitrile Content on the Toughness of ABS Materials’. In: *Polymer* 32.8 (Jan. 1991), pp. 1447–1455. DOI: 10.1016/0032-3861(91)90425-I.
- [18] P. Fratzl and R. Weinkamer. ‘Nature’s Hierarchical Materials’. In: *Progress in Materials Science* 52.8 (Nov. 2007), pp. 1263–1334. DOI: 10.1016/j.pmatsci.2007.06.001.
- [19] F. Barthelat and H. D. Espinosa. ‘An Experimental Investigation of Deformation and Fracture of Nacre–Mother of Pearl’. In: *Experimental Mechanics* 47.3 (June 2007), pp. 311–324. DOI: 10.1007/s11340-007-9040-1.
- [20] M. Cervera and M. Chiumenti. ‘Smearred Crack Approach: Back to the Original Track’. In: *International Journal for Numerical and Analytical Methods in Geomechanics* 30.12 (Oct. 2006), pp. 1173–1199. DOI: 10.1002/nag.518.
- [21] N. Moës, J. Dolbow and T. Belytschko. ‘A Finite Element Method for Crack Growth without Remeshing’. In: *International Journal for Numerical Methods in Engineering* 46.1 (Sept. 1999), pp. 131–150. DOI: 10.1002/(SICI)1097-0207(19990910)46:1<131::AID-NME726>3.0.CO;2-J.



- 
- [22] T. Belytschko and T. Black. ‘Elastic Crack Growth in Finite Elements with Minimal Remeshing’. In: *International Journal for Numerical Methods in Engineering* 45.5 (June 1999), pp. 601–620. DOI: 10.1002/(SICI)1097-0207(19990620)45:5<601::AID-NME598>3.0.CO;2-S.
- [23] H. Li, J. Li and H. Yuan. ‘A Review of the Extended Finite Element Method on Macrocrack and Microcrack Growth Simulations’. In: *Theoretical and Applied Fracture Mechanics* 97 (Oct. 2018), pp. 236–249. DOI: 10.1016/j.tafmec.2018.08.008.
- [24] Z. P. Bažant and B. H. Oh. ‘Crack Band Theory for Fracture of Concrete’. In: *Matériaux et Constructions* 16.3 (May 1983), pp. 155–177. DOI: 10.1007/BF02486267.
- [25] R. Peerlings, M. Geers, R. De Borst and W. Brekelmans. ‘A Critical Comparison of Nonlocal and Gradient-Enhanced Softening Continua’. In: *International Journal of Solids and Structures* 38.44-45 (Nov. 2001), pp. 7723–7746. DOI: 10.1016/S0020-7683(01)00087-7.
- [26] T. Q. Bui and X. Hu. ‘A Review of Phase-Field Models, Fundamentals and Their Applications to Composite Laminates’. In: *Engineering Fracture Mechanics* 248 (May 2021), p. 107705. DOI: 10.1016/j.engfracmech.2021.107705.
- [27] C. Miehe, M. Hofacker and F. Welschinger. ‘A Phase Field Model for Rate-Independent Crack Propagation: Robust Algorithmic Implementation Based on Operator Splits’. In: *Computer Methods in Applied Mechanics and Engineering* 199.45-48 (Nov. 2010), pp. 2765–2778. DOI: 10.1016/j.cma.2010.04.011.
- [28] E. Bormetti, G. Donzella and A. Mazzù. ‘Surface and Subsurface Cracks in Rolling Contact Fatigue of Hardened Components’. In: *Tribology Transactions* 45.3 (Jan. 2002), pp. 274–283. DOI: 10.1080/10402000208982550.
- [29] M. Torabizadeh, Z. A. Putnam, M. Sankarasubramanian, J. C. Moosbrugger and S. Krishnan. ‘The Effects of Initial Crack Length on Fracture Characterization of Rubbers Using the J-Integral Approach’. In: *Polymer Testing* 73 (Feb. 2019), pp. 327–337. DOI: 10.1016/j.polymertesting.2018.11.026.
- [30] R. v. Mises. ‘Mechanik Der Festen Körper Im Plastisch- Deformablen Zustand’. In: *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* 1913 (1913), pp. 582–592.
- [31] P. Cornetti, N. Pugno, A. Carpinteri and D. Taylor. ‘Finite Fracture Mechanics: A Coupled Stress and Energy Failure Criterion’. In: *Engineering Fracture Mechanics* 73.14 (Sept. 2006), pp. 2021–2033. DOI: 10.1016/j.engfracmech.2006.03.010.
- [32] R. M. Christensen. *The Theory of Materials Failure*. 1st ed. Oxford: Oxford University Press, 2013.

- [33] J. Li and D. Leguillon. ‘Finite Element Implementation of the Coupled Criterion for Numerical Simulations of Crack Initiation and Propagation in Brittle Materials’. In: *Theoretical and Applied Fracture Mechanics* 93 (Feb. 2018), pp. 105–115. DOI: 10.1016/j.tafmec.2017.07.010.
- [34] M. Rettl, M. Pletz and C. Schuecker. *Optimizing the Hole Geometry of 2d Plates for Maximum Tensile Toughness*. 2024. DOI: 10.2139/ssrn.4962118.
- [35] J. R. Lund and J. P. Byrne. ‘Leonardo Da Vinci’s Tensile Strength Test: Implications for the Discovery of Engineering Mechanics’. In: *Civil Engineering and Environmental Systems* 18.3 (June 2001), pp. 243–250. DOI: 10.1080/02630250108970302.
- [36] A. A. Griffith. ‘VI. The Phenomena of Rupture and Flow in Solids’. In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 221.582-593 (Jan. 1921), pp. 163–198. DOI: 10.1098/rsta.1921.0006.
- [37] C. E. Inglis. ‘Stresses in a Plate Due to the Presence of Cracks and Sharp Corners’. In: *Transactions of the Institute of Naval Architects* 55 (1913), pp. 219–241.
- [38] H. Tada, P. C. Paris and G. R. Irwin. *The Stress Analysis of Cracks Handbook*. St. Louis: Paris Productions, Inc., 1985.
- [39] H. Westergaard. ‘Bearing Pressures and Cracks’. In: *Journal of Applied Mechanics* 6 (1939), pp. 49–53.
- [40] G. R. Irwin. ‘Analysis of Stresses and Strains Near the End of a Crack Traversing a Plate’. In: *Journal of Applied Mechanics* 24.3 (Sept. 1957), pp. 361–364. DOI: 10.1115/1.4011547.
- [41] M. L. Williams. ‘On the Stress Distribution at the Base of a Stationary Crack’. In: *Journal of Applied Mechanics* 24.1 (Mar. 1957), pp. 109–114. DOI: 10.1115/1.4011454.
- [42] Y. Wang, W. Wang, B. Zhang and C.-Q. Li. ‘A Review on Mixed Mode Fracture of Metals’. In: *Engineering Fracture Mechanics* 235 (Aug. 2020), p. 107126. DOI: 10.1016/j.engfracmech.2020.107126.
- [43] W. Hua, J. Li, Z. Zhu, A. Li, J. Huang, Z. Gan and S. Dong. ‘A Review of Mixed Mode I-II Fracture Criteria and Their Applications in Brittle or Quasi-Brittle Fracture Analysis’. In: *Theoretical and Applied Fracture Mechanics* 124 (Apr. 2023), p. 103741. DOI: 10.1016/j.tafmec.2022.103741.
- [44] F. Erdogan and G. C. Sih. ‘On the Crack Extension in Plates Under Plane Loading and Transverse Shear’. In: *Journal of Basic Engineering* 85.4 (Dec. 1963), pp. 519–525. DOI: 10.1115/1.3656897.

- 
- [45] G. C. Sih. ‘Strain-Energy-Density Factor Applied to Mixed Mode Crack Problems’. In: *International Journal of Fracture* 10.3 (Sept. 1974), pp. 305–321. DOI: 10.1007/BF00035493.
- [46] A. Boulenouar, N. Benseddiq, M. Merzoug, N. Benamara and M. Mazari. ‘A Strain Energy Density Theory for Mixed Mode Crack Propagation in Rubber-like Materials’. In: *Journal of Theoretical and Applied Mechanics* (Oct. 2016), p. 1417. DOI: 10.15632/jtam-pl.54.4.1417.
- [47] M. Hussain, S. Pu and J. Underwood. ‘Strain Energy Release Rate for a Crack Under Combined Mode I and Mode II’. In: *National Symposium on Fracture Mechanics*. Ed. by G. Irwin. 100 Barr Harbor Drive, PO Box C700, West Conshohocken, PA 19428-2959: ASTM International, Jan. 1974, pp. 2-2–27. DOI: 10.1520/STP33130S.
- [48] J. R. Rice. ‘A Path Independent Integral and the Approximate Analysis of Strain Concentration by Notches and Cracks’. In: *Journal of Applied Mechanics* 35.2 (June 1968), pp. 379–386. DOI: 10.1115/1.3601206.
- [49] B. Budiansky and J. R. Rice. ‘Conservation Laws and Energy-Release Rates’. In: *Journal of Applied Mechanics* 40.1 (Mar. 1973), pp. 201–203. DOI: 10.1115/1.3422926.
- [50] P. O. Judt and A. Ricoeur. ‘Crack Growth Simulation of Multiple Cracks Systems Applying Remote Contour Interaction Integrals’. In: *Theoretical and Applied Fracture Mechanics* 75 (Feb. 2015), pp. 78–88. DOI: 10.1016/j.tafmec.2014.11.001.
- [51] S. M. Frankl, M. Pletz and C. Schuecker. ‘Improved Concept for Iterative Crack Propagation Using Configurational Forces for Targeted Angle Correction’. In: *Engineering Fracture Mechanics* 266 (May 2022), p. 108403. DOI: 10.1016/j.engfracmech.2022.108403.
- [52] M. Smith. *ABAQUS/Standard User’s Manual, Version 6.9*. United States: Dassault Systèmes Simulia Corp, 2009.
- [53] D. Parks. ‘The Virtual Crack Extension Method for Nonlinear Material Behavior’. In: *Computer Methods in Applied Mechanics and Engineering* 12.3 (Dec. 1977), pp. 353–364. DOI: 10.1016/0045-7825(77)90023-8.
- [54] M. E. Gurtin. *Configurational Forces as Basic Concepts of Continuum Physics*. Vol. 137. Applied Mathematical Sciences. New York, NY: Springer New York, 2000. DOI: 10.1007/b97847.
- [55] J. D. Eshelby. ‘The Elastic Energy-Momentum Tensor’. In: *Journal of Elasticity* 5.3-4 (Nov. 1975), pp. 321–335. DOI: 10.1007/BF00126994.
- [56] R. Mueller and G. A. Maugin. ‘On Material Forces and Finite Element Discretizations’. In: *Computational Mechanics* 29.1 (July 2002), pp. 52–60. DOI: 10.1007/s00466-002-0322-2.

- [57] K. Schmitz and A. Ricoeur. ‘Theoretical and Computational Aspects of Configurational Forces in Three-Dimensional Crack Problems’. In: *International Journal of Solids and Structures* 282 (Oct. 2023), p. 112456. DOI: 10.1016/j.ijsolstr.2023.112456.
- [58] R. Denzer, F. J. Barth and P. Steinmann. ‘Studies in Elastic Fracture Mechanics Based on the Material Force Method’. In: *International Journal for Numerical Methods in Engineering* 58.12 (Nov. 2003), pp. 1817–1835. DOI: 10.1002/nme.834.
- [59] M. Rettl, S. Frankl, M. Pletz, M. Tauscher and C. Schuecker. ‘ConForce: Computation of Configurational Forces for FEM Results’. In: *SoftwareX* 26 (May 2024), p. 101718. DOI: 10.1016/j.softx.2024.101718.
- [60] Z. Hashin. ‘Finite Thermoelastic Fracture Criterion with Application to Laminate Cracking Analysis’. In: *Journal of the Mechanics and Physics of Solids* 44.7 (July 1996), pp. 1129–1145. DOI: 10.1016/0022-5096(95)00080-1.
- [61] D. Taylor, P. Cornetti and N. Pugno. ‘The Fracture Mechanics of Finite Crack Extension’. In: *Engineering Fracture Mechanics* 72.7 (May 2005), pp. 1021–1038. DOI: 10.1016/j.engfracmech.2004.07.001.
- [62] D. Taylor. ‘The Theory of Critical Distances’. In: *Engineering Fracture Mechanics* 75.7 (May 2008), pp. 1696–1705. DOI: 10.1016/j.engfracmech.2007.04.007.
- [63] D. Leguillon. ‘Strength or Toughness? A Criterion for Crack Onset at a Notch’. In: *European Journal of Mechanics - A/Solids* 21.1 (Jan. 2002), pp. 61–72. DOI: 10.1016/S0997-7538(01)01184-6.
- [64] D. Leguillon and Z. Yosibash. ‘Crack Onset at a V-Notch. Influence of the Notch Tip Radius’. In: *International Journal of Fracture* 122.1/2 (July 2003), pp. 1–21. DOI: 10.1023/B:FRAC.0000005372.68959.1d.
- [65] D. Taylor. ‘Geometrical Effects in Fatigue: A Unifying Theoretical Model’. In: *International Journal of Fatigue* 21.5 (May 1999), pp. 413–420. DOI: 10.1016/S0142-1123(99)00007-9.
- [66] D. Taylor. ‘Predicting the Fracture Strength of Ceramic Materials Using the Theory of Critical Distances’. In: *Engineering Fracture Mechanics* 71.16-17 (Nov. 2004), pp. 2407–2416. DOI: 10.1016/j.engfracmech.2004.01.002.
- [67] V. Madrazo, S. Cicero and I. Carrascal. ‘On the Point Method and the Line Method Notch Effect Predictions in Al7075-T651’. In: *Engineering Fracture Mechanics* 79 (Jan. 2012), pp. 363–379. DOI: 10.1016/j.engfracmech.2011.11.017.
- [68] P. Weißgraeber, D. Leguillon and W. Becker. ‘A Review of Finite Fracture Mechanics: Crack Initiation at Singular and Non-Singular Stress Raisers’. In: *Archive of Applied Mechanics* 86.1-2 (Jan. 2016), pp. 375–401. DOI: 10.1007/s00419-015-1091-7.

- 
- [69] A. Doitrand, T. Duminy, H. Girard and X. Chen. ‘A Review of the Coupled Criterion’. In: (2023).
- [70] A. Carpinteri, P. Cornetti, N. Pugno, A. Sapora and D. Taylor. ‘A Finite Fracture Mechanics Approach to Structures with Sharp V-notches’. In: *Engineering Fracture Mechanics* 75.7 (May 2008), pp. 1736–1752. DOI: 10.1016/j.engfracmech.2007.04.010.
- [71] A. Carpinteri, P. Cornetti and A. Sapora. ‘A Finite Fracture Mechanics Approach to the Asymptotic Behaviour of U-notched Structures’. In: *Fatigue & Fracture of Engineering Materials & Structures* 35.5 (May 2012), pp. 451–457. DOI: 10.1111/j.1460-2695.2011.01637.x.
- [72] A. Sapora, P. Cornetti and A. Carpinteri. ‘A Finite Fracture Mechanics Approach to V-notched Elements Subjected to Mixed-Mode Loading’. In: *Engineering Fracture Mechanics* 97 (Jan. 2013), pp. 216–226. DOI: 10.1016/j.engfracmech.2012.11.006.
- [73] R. Romani, M. Bornert, D. Leguillon, R. Le Roy and K. Sab. ‘Detection of Crack Onset in Double Cleavage Drilled Specimens of Plaster under Compression by Digital Image Correlation – Theoretical Predictions Based on a Coupled Criterion’. In: *European Journal of Mechanics - A/Solids* 51 (May 2015), pp. 172–182. DOI: 10.1016/j.euromechsol.2014.12.002.
- [74] A. Doitrand, E. Martin and D. Leguillon. ‘Numerical Implementation of the Coupled Criterion: Matched Asymptotic and Full Finite Element Approaches’. In: *Finite Elements in Analysis and Design* 168 (Jan. 2020), p. 103344. DOI: 10.1016/j.finel.2019.103344.
- [75] Z. Yosibash, E. Priel and D. Leguillon. ‘A Failure Criterion for Brittle Elastic Materials under Mixed-Mode Loading’. In: *International Journal of Fracture* 141.1-2 (Sept. 2006), pp. 291–312. DOI: 10.1007/s10704-006-0083-6.
- [76] D. Leguillon, D. Quesada, C. Putot and E. Martin. ‘Prediction of Crack Initiation at Blunt Notches and Cavities – Size Effects’. In: *Engineering Fracture Mechanics* 74.15 (Oct. 2007), pp. 2420–2436. DOI: 10.1016/j.engfracmech.2006.11.008.
- [77] A. Doitrand, R. Estevez and D. Leguillon. ‘Comparison between Cohesive Zone and Coupled Criterion Modeling of Crack Initiation in Rhombus Hole Specimens under Quasi-Static Compression’. In: *Theoretical and Applied Fracture Mechanics* 99 (Feb. 2019), pp. 51–59. DOI: 10.1016/j.tafmec.2018.11.007.
- [78] A. Sapora, A. Torabi, S. Etesam and P. Cornetti. ‘Finite Fracture Mechanics Crack Initiation from a Circular Hole’. In: *Fatigue & Fracture of Engineering Materials & Structures* 41.7 (July 2018), pp. 1627–1636. DOI: 10.1111/ffe.12801.

- [79] A. Sapora, G. Efremidis and P. Cornetti. ‘Comparison between Two Nonlocal Criteria: A Case Study on Pressurized Holes’. In: *Procedia Structural Integrity* 33 (2021), pp. 456–464. DOI: 10.1016/j.prostr.2021.10.052.
- [80] D. Leguillon, E. Martin, O. Ševeček and R. Bermejo. ‘Application of the Coupled Stress-Energy Criterion to Predict the Fracture Behaviour of Layered Ceramics Designed with Internal Compressive Stresses’. In: *European Journal of Mechanics - A/Solids* 54 (Nov. 2015), pp. 94–104. DOI: 10.1016/j.euromechsol.2015.06.008.
- [81] O. Ševeček, M. Kotoul, D. Leguillon, E. Martin and R. Bermejo. ‘Assessment of Crack-Related Problems in Layered Ceramics Using the Finite Fracture Mechanics and Coupled Stress-Energy Criterion’. In: *Procedia Structural Integrity* 2 (2016), pp. 2014–2021. DOI: 10.1016/j.prostr.2016.06.253.
- [82] A. Parvizi, K. W. Garrett and J. E. Bailey. ‘Constrained Cracking in Glass Fibre-Reinforced Epoxy Cross-Ply Laminates’. In: *Journal of Materials Science* 13.1 (Jan. 1978), pp. 195–201. DOI: 10.1007/BF00739291.
- [83] M. Rettl, M. Pletz and C. Schuecker. ‘Efficient Prediction of Crack Initiation from Arbitrary 2D Notches’. In: *Theoretical and Applied Fracture Mechanics* 119 (June 2022), p. 103376. DOI: 10.1016/j.tafmec.2022.103376.
- [84] M. Rettl. ‘Effiziente Vorhersage der Rissinitiierung in porösen Materialien mit einem kombinierten Energie- und Spannungskriterium’. MA thesis. Montanuniversität Leoben, 2020.
- [85] E. Buckingham. ‘On Physically Similar Systems; Illustrations of the Use of Dimensional Equations’. In: *Physical Review* 4.4 (Oct. 1914), pp. 345–376. DOI: 10.1103/PhysRev.4.345.
- [86] R. Damani, R. Gstrein and R. Danzer. ‘Critical Notch-Root Radius Effect in SENB-S Fracture Toughness Testing’. In: *Journal of the European Ceramic Society* 16.7 (Jan. 1996), pp. 695–702. DOI: 10.1016/0955-2219(95)00197-2.
- [87] M. Bijak-Zochowski, A. M. Waas, W. J. Anderson and C. E. Miniatt. ‘Reduction of Contact Stress by Use of Relief Notches’. In: *Experimental Mechanics* 31.3 (Sept. 1991), pp. 271–275. DOI: 10.1007/BF02326071.
- [88] M. Rettl, M. Pletz and C. Schuecker. ‘Evaluation of Combinatorial Algorithms for Optimizing Highly Nonlinear Structural Problems’. In: *Materials & Design* 230 (June 2023), p. 111958. DOI: 10.1016/j.matdes.2023.111958.
- [89] O. Sigmund and K. Maute. ‘Topology Optimization Approaches: A Comparative Review’. In: *Structural and Multidisciplinary Optimization* 48.6 (Dec. 2013), pp. 1031–1055. DOI: 10.1007/s00158-013-0978-6.

- 
- [90] N. Mladenović and P. Hansen. ‘Variable Neighborhood Search’. In: *Computers & Operations Research* 24.11 (Nov. 1997), pp. 1097–1100. DOI: 10.1016/S0305-0548(97)00031-2.
- [91] P. Hansen, N. Mladenović and J. A. Moreno Pérez. ‘Variable Neighbourhood Search: Methods and Applications’. In: *Annals of Operations Research* 175.1 (Mar. 2010), pp. 367–407. DOI: 10.1007/s10479-009-0657-6.
- [92] N. Mladenovic, A. Sleptchenko, A. Sifaleras and M. Omar. *Variable Neighborhood Search: 8th International Conference, ICVNS 2021, Abu Dhabi, United Arab Emirates, March 21–25, 2021, Proceedings*. Vol. 12559. Springer Nature, 2021.
- [93] C. Kane and M. Schoenauer. ‘Topological Optimum Design Using Genetic Algorithms’. In: *Control and Cybernetics* 25.5 (1996), pp. 1059–1088.
- [94] M. J. Jakiela, C. Chapman, J. Duda, A. Adewuya and K. Saitou. ‘Continuum Structural Topology Design with Genetic Algorithms’. In: *Computer Methods in Applied Mechanics and Engineering* 186.2-4 (June 2000), pp. 339–356. DOI: 10.1016/S0045-7825(99)00390-4.
- [95] R. Balamurugan, C. Ramakrishnan and N. Singh. ‘Performance Evaluation of a Two Stage Adaptive Genetic Algorithm (TSAGA) in Structural Topology Optimization’. In: *Applied Soft Computing* 8.4 (Sept. 2008), pp. 1607–1624. DOI: 10.1016/j.asoc.2007.10.022.
- [96] R. Balamurugan, C. V. Ramakrishnan and N. Swaminathan. ‘A Two Phase Approach Based on Skeleton Convergence and Geometric Variables for Topology Optimization Using Genetic Algorithm’. In: *Structural and Multidisciplinary Optimization* 43.3 (Mar. 2011), pp. 381–404. DOI: 10.1007/s00158-010-0560-4.
- [97] P. Y. Shim and S. Manoochchri. ‘Generating Optimal Configurations in Structural Design Using Simulated Annealing’. In: *International Journal for Numerical Methods in Engineering* 40.6 (Mar. 1997), pp. 1053–1069. DOI: 10.1002/(SICI)1097-0207(19970330)40:6<1053::AID-NME97>3.0.CO;2-I.
- [98] M. Kanagawa, P. Hennig, D. Sejdinovic and B. K. Sriperumbudur. *Gaussian Processes and Kernel Methods: A Review on Connections and Equivalences*. July 2018. arXiv: 1807.02582 [cs, stat].
- [99] R. L. Hardy. ‘Multiquadric Equations of Topography and Other Irregular Surfaces’. In: *Journal of Geophysical Research* 76.8 (Mar. 1971), pp. 1905–1915. DOI: 10.1029/JB076i008p01905.





# Papers



**Paper A**

# Efficient Prediction of Crack Initiation from Arbitrary 2D Notches

Authors: Rettl, Matthias  
Pletz, Martin  
Schuecker, Clara

Theoretical and Applied Fracture Mechanics

DOI: [10.1016/j.tafmec.2022.103376](https://doi.org/10.1016/j.tafmec.2022.103376)

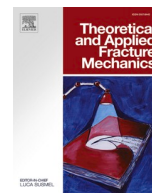




Contents lists available at ScienceDirect

# Theoretical and Applied Fracture Mechanics

journal homepage: [www.elsevier.com/locate/tafmec](http://www.elsevier.com/locate/tafmec)



## Efficient prediction of crack initiation from arbitrary 2D notches

Matthias Retzl, Martin Pletzl\*, Clara Schuecker

Designing Plastics and Composite Materials, Department of Polymer Engineering and Science, Montanuniversitaet Leoben, Austria

### ARTICLE INFO

#### Keywords:

Coupled criterion  
Finite element analysis  
Fracture mechanics  
Meta-modeling

### ABSTRACT

An efficient two-scale approach for predicting mode I crack initiation from 2D notches based on the Coupled Criterion is proposed. On the scale of the *local model*, a voxel model containing the notch simulates the displacement field. The *crack model* is introduced on the smaller scale and is defined in an image space. Based on the notch curvature, the precomputed crack model can be transformed to any position on the notch surface. The displacement field of the local model is fitted at the boundaries of the transformed crack model by predefined deformation modes and results can be obtained by a superposition of precomputed crack model results. By introducing the crack in the crack model, the stiffness of this model is reduced and thus, the incremental energy release rate can be inaccurate. Therefore, a boundary relaxation approach is used to obtain more accurate energy release rates. It is shown that the method is very efficient as it requires only 3:20 min to analyze 50 positions on a notch compared to 2:21 h of a conventional approach using full FEM simulations. Thereby, the method is reliable in identifying the critical position. The predicted failure index at this position deviates by at most 10.8%. Since the crack model limits the length of initiating cracks, Irwin's length  $K_{Ic}^2/\sigma_c^2$  of the material must lie below 2.53 times the radius of a circular hole under uniaxial tension. For a brittle material like  $Al_2O_3$ , notches with a curvature radius above  $31\mu m$  can thus be analyzed.

### 1. Introduction

Notches are inevitable in engineering components and often lead to failure. Failure is commonly predicted using stress-based approaches. Such criteria like the maximum principal stress criterion define a material-dependent failure surface that must not be exceeded by the computed stresses, otherwise, failure will occur [1]. The stresses are computed for each position to check if a component survives the applied load. For the computation of an arbitrary-shaped component, only one FEA simulation is required. This makes these approaches versatile and efficient, but they lack accuracy for some cases. For sharp cracks, the computed stress at the crack tip rises to infinity regardless of the applied load and each stress-based criterion immediately predicts failure, whereas experiments show that even in the vicinity of cracks, failure doesn't occur until a critical load is applied [2].

The Theory of Critical Distances (TCD) honors this finding by averaging the stress from the surface of the notch to a critical distance into the material. The averaged stress must not exceed the strength, otherwise, failure will occur. The critical distance is assumed to be a material parameter [3]. Unfortunately, this assumption is not plausible, as Taylor demonstrated with the example of a bending beam [4].

In contrast to defining the critical distance as a material parameter, Leguillon's Coupled Criterion (CC) computes the critical distance by combining a stress- and an energy-based criterion [5]. The CC within the finite fracture mechanics framework investigates the instantaneous initiation of a finite size crack [6]. Experiments with V-notched bending beams of PMMA, PS, or brittle ceramics show good agreement with the CC [7–10], whereas uniaxial tensile tests on PMMA plates containing a hole are predicted mediocly by the CC [11]. However, compression tests on plates with a hole show a relative error between the CC predictions and the average experimental applied stress of at most 20% [12].

Currently, there are two possibilities to implement CC. The first one, called Matched Asymptotic approach (MA), is extensively discussed in the literature for sharp and blunted V-notches [5,7,8,13,14] and plates containing a circular hole [13,15]. The MA approach is a two-scale approach with a far and a near field scale. It is an efficient and accurate method, but it is limited to relatively simple geometries [12] and is not versatile enough for engineering problems. The second method discretizes the geometry using the Finite Element Analysis. We call this method *Full FEA*. It is indeed versatile and accurate enough to handle complicated geometries, but several FEA computations are necessary that significantly increase the computation time, especially if Full FEA

\* Corresponding author.

E-mail address: [martin.pletzl@unileoben.ac.at](mailto:martin.pletzl@unileoben.ac.at) (M. Pletzl).

<https://doi.org/10.1016/j.tafmec.2022.103376>

Received 14 January 2022; Received in revised form 29 March 2022; Accepted 25 April 2022

Available online 30 April 2022

0167-8442/© 2022 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Nomenclature		Superscripts	
<b>Acronyms</b>		$Q$	quantity in real space
TCD	Theory of Critical Distances	$Q'$	quantity in virtual space
CC	Coupled Criterion	<b>Parameters</b>	
MA	Matched Asymptotic	$\vec{\Gamma}$	notch surface
FEA (FEM)	Finite Element Analysis (Finite Element Method)	$s$	position on the notch surface
SLMM	Scaling Law Meta-Model	$c$	curvature of the notch surface
DC	Displacement Control	$a$	virtual crack length
FC	Force Control	$a_0, \dots, a_n$	introduced virtual crack lengths
AC	Auto Control	$l, h, b, r$	length, height, thickness, radius
$Al_2O_3$	Aluminium oxide	$l_{ch}$	Irwin's length
PMMA	Poly(methyl methacrylate)	$\varphi_{min}$	ratio $h_{cm}$ to $l_{el,lm}$
PS	Polystyrene	$\rho, \rho_{solid}, \rho_{notch}$	element densities
<b>Coordinate systems</b>		$E$	Young's modulus
xy	specimen coordinate system	$\nu$	Poisson's ratio
$\xi\eta$	aligned normal to the notch surface	$\vec{u}, \vec{u}_{off}$	displacement vector
$\xi'\eta'$	virtual space coordinate system	$u_{0,i}$	reference value of a deformation mode
<b>Matrix and vector convention</b>		$\mathbb{D}, \mathbb{D}_{all}, \mathbb{D}_{imp}$	set of deformation modes
$\vec{Q} = (Q_i)_i$	vector and summation notation	$\vec{F}$	reaction force
$\mathbf{Q} = (Q_{ik})_{i,k}$	matrix and summation notation	$\mathbf{K}$	stiffness matrix
<b>Subscripts</b>		$\Psi$	difference of stiffness matrices
$Q_c$	critical quantity	$\sigma$	stress
$Q_{up}$	upper bound for $Q_c$	$\sigma_c$	material strength
$Q_{low}$	lower bound for $Q_c$	$\varepsilon, \gamma$	normal strain and shear strain
$Q_{max}$	maximum quantity	$\Pi$	strain energy
$Q_{min}$	maximum quantity	$G_{inc}$	incremental energy release rate
$Q_{xx}, Q_{xy}, Q_{yy}$	tensor components in xy	$G_c$	material fracture energy
$Q_{\xi\xi}, Q_{\xi\eta}, Q_{\eta\eta}$	tensor components in $\xi\eta$	$K_{Ic}$	material toughness
$Q_I$	maximum principal	$lf, FI$	load factor and failure index
$Q_{FC}$	force-controlled quantity	$\lambda_\sigma, \lambda_l$	scaling variables
$Q_{MC}$	mixed-controlled quantity	$n_{cracks}$	number of introduced virtual crack lengths
$Q_{el}$	corresponds to elements	$n_{fit,c}, n_{fit,u}$	number of sampling points
$Q_{full}$	corresponds to the full FEM	$n_{pos}$	number of positions on the notch surface
$Q_{lm}$	corresponds to the local model	$n_{el,lm}$	number of voxel elements
$Q_{cm}$	corresponds to the crack model	$\Delta_{abs}$	absolute tolerance
$Q_{RP}$	corresponds to a reference point	$\Delta_{rel}$	relative tolerance
		$\Delta G_{inc}, \Delta\sigma_I$	maximum relative effect by changing a parameter
		$ApI_{max}$	applicability factor

must be applied to several candidate positions, because the critical position on a notch is not known. Li [16] uses a sophisticated watershed floating process to find critical positions, but the method is still not efficient enough for engineering problems. Furthermore, cohesive zone models (CZM) provide close predictions to the CC but need computationally expensive non-linear analyses and need to be manually placed on critical positions on the notch [17–19]. For engineering problems, stress-based or TCD approaches might be inaccurate in some cases. The CC on the other side agrees well with experiments, but its implementation lacks either in versatility when using MA or in efficiency when using Full FEA. CZM are not practical for arbitrary shaped notches if the crack path is not known a priori.

The goal of this work is to propose a novel implementation of CC that is efficient, versatile, and accurate enough for engineering problems. Therefore, we use FEA computations, as with the Full FEA approach, and a multi-scale model, as with MA. We use two FEA models. The *local model* contains the whole notch and is implemented as a voxel-model, because it is easy to create a voxel-model from an image of a notch. The *crack model* contains the crack and predicts the stresses and the incremental energy release rate needed by the CC. The crack model is used like a submodel of the local model. One drawback of using submodels is the assumption that the crack is small compared to the size of

the submodel [20]. Therefore, we propose a relaxation mechanism for the boundary conditions of the crack model that weakens the assumptions and gives accurate predictions. Furthermore, the crack model is precomputed in advance for a given material. Like Budinger [21], we utilize a scaling law and a *meta*-model. This allows us to apply the precomputed model to arbitrary notch geometries efficiently and versatilely.

Crack initiation in brittle materials often leads to unstable crack propagation and component failure. To state whether crack initiation and subsequent failure occurs, we define a failure index that predicts crack initiation if it becomes 1 or higher. We validate the proposed method with the Full FEA approach and show that the most critical failure indices for four examples deviate at most 10.8%. However, the method needs only a few minutes to compute the failure index along a whole notch surface, whereas the Full FEA takes several hours to do so.

## 2. Methods

The aim of this work is to provide efficient, versatile, and accurate predictions of crack initiation at a notch using the Coupled Criterion (CC). For that purpose, the maximum principal stress  $\sigma_I(\eta)$  along a virtual crack path  $\eta$  and the strain energy  $\Pi(a)$  for the virtual crack with

various crack lengths  $a$  is required. Fig. 1 depicts the two possibilities Full FEA and our novel approach SLMM to calculate  $\sigma_1(\eta)$  and  $\Pi(a)$  at a position  $s = s_p$  on the notch. Both methods start with a model containing a notch. For example, a strain tensor can be used as a boundary condition at the margins of the notch model. The surface of the notch is described as a cubic B-spline.

$$\vec{\Gamma}(s) = \begin{pmatrix} \Gamma_x(s) \\ \Gamma_y(s) \end{pmatrix} \quad (1)$$

where  $s$  defines the position on the spline [22]. The position  $s$  is defined as the path length from the starting point. The first approach Full FEA is described in detail in section 2.2.1 and uses one FEM analysis to compute the max. principal stress  $\sigma_1(\eta)$  and the strain energy  $\Pi_0$ . Further analyses are necessary with various introduced crack lengths  $a_i$  to compute the strain energies  $\Pi(a_i) = \Pi_i$ . The second novel approach SLMM uses a voxel-model to compute the displacement field  $\vec{u}(x,y)$ . The curvature  $c(s)$  is derived from the spline  $\vec{\Gamma}(s)$ . These parameters are transformed into image space. Quantities in the image space are marked by an apostrophe. Next, a precomputed crack model in the image space predicts the stress tensor  $\sigma'(\eta')$  as well as the strain energy  $\Pi'(a')$ . After the back transformation to the real space,  $\sigma_1(\eta)$  and  $\Pi(a)$  are computed.

2.1. Coupled criterion (CC)

The CC states whether a crack initiates at a specific position  $P$  of a notch under a certain load. Therefore, CC needs the strength  $\sigma_c$  and fracture energy  $G_c$  of the material as well as the computed stress and strain energy curves along a virtual crack path  $\eta$ . As shown in Fig. 2a-c), we assume a crack path  $\eta$  normal to the notch surface straight into the material and we use the maximum principal stress  $\sigma_1(\eta)$  that is evaluated in a model without a crack. In contrast, the strain energies  $\Pi_i$  are computed in models containing a virtual crack with lengths  $a_i$ . The strain energy  $\Pi(a)$  is interpolated piecewise linearly.

The incremental energy release rate.

$$G_{inc}(a) := \begin{cases} \frac{\Pi_0 - \Pi(a)}{b \cdot a} & \text{if } a > 0 \\ 0 \frac{J}{m^2} & \text{if } a = 0 \end{cases} \quad (2)$$

is the difference of the strain energy without a crack  $\Pi_0$  and with a virtual crack  $\Pi(a)$  in relation to the area of the virtual crack surface  $b \cdot a$ , where  $b$  is the thickness of the specimen and  $a$  is the length of the virtual crack. According to the energy criterion depicted in Fig. 3a), initiation is possible for a length  $a_i$  and above where  $G_{inc}$  is higher than the fracture energy  $G_c$ . The energy criterion thus results in a lower bound  $a_i$  for the crack length  $a_c$ .

On the other hand, the stress-criterion depicted in Fig. 3b) states that initiation is only possible up to a length  $a_u$ , where the max. principal stress  $\sigma_1$  is higher than the strength  $\sigma_c$ . Consequently, the stress criterion gives an upper bound for the crack length  $a_u$ .

In linear analyses, a load factor  $lf \geq 0$  can be used to scale the load linearly. The max. principal stress  $\sigma_1 \propto lf$  is proportional to the load factor, whereas  $G_{inc} \propto lf^2$  is scaled quadratically. The CC shown in Fig. 3c) computes the critical load factor  $lf_c$  such that the energy- and stress-criterion are fulfilled simultaneously at the crack length  $a_c$ . Furthermore, a failure index  $FI = 1/lf_c$  is defined [23]. There are three possible statements:

1.  $FI \leq 0$ : No crack will initiate.
2.  $0 < FI < 1$ : No crack initiates under the current load, but if the load is scaled by  $lf_c$  initiation will occur.
3.  $1 \leq FI$ : A crack will initiate under the current load.

2.2. FEM models

This section describes the used FEM models. The Full FEA approach computes the max. principal stress  $\sigma_1(\eta)$  and the incremental energy release rate  $G_{inc}(a)$  directly, whereas the local model and the crack model are used within the novel SLMM approach to compute  $\sigma_1(\eta)$  and  $G_{inc}(a)$ .

2.2.1. Full FEA

Full FEA is a common approach to implement CC. As depicted in Fig. 1, Full FEA uses fully modeled FEM analyses. The models contain a virtual crack with increasing length  $a$ . In total,  $1 + n_{cracks}$  virtual crack lengths  $a_i$  are introduced from  $a_0 = 0\text{mm}$  to  $a_n$ . We chose the maximum crack length  $a_n$  as described in section 2.3. In each simulation, the strain energy  $\Pi_i$  is evaluated. In the first simulation with  $a = a_0$ , the max.

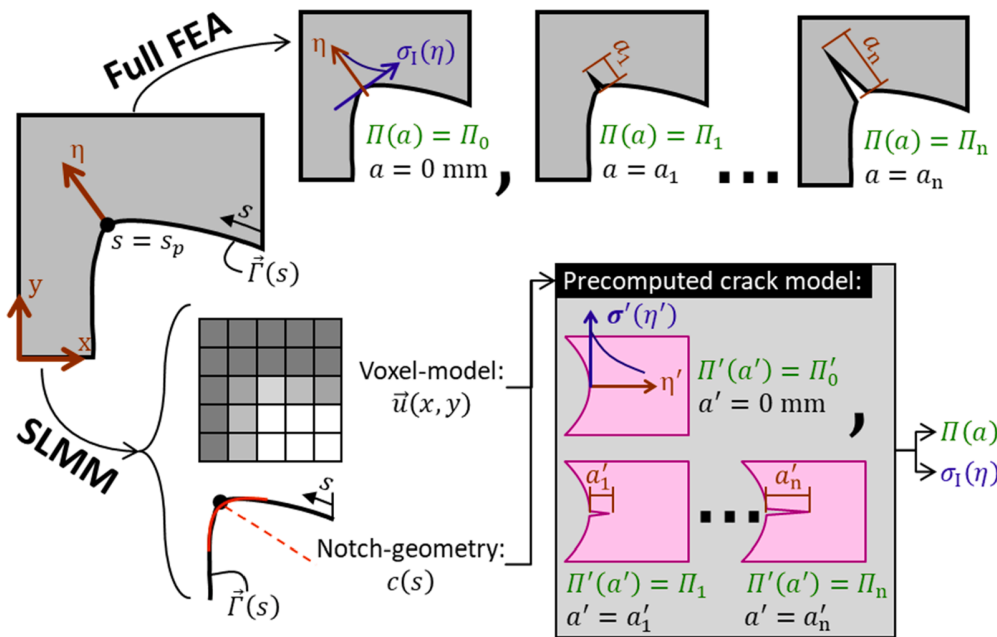
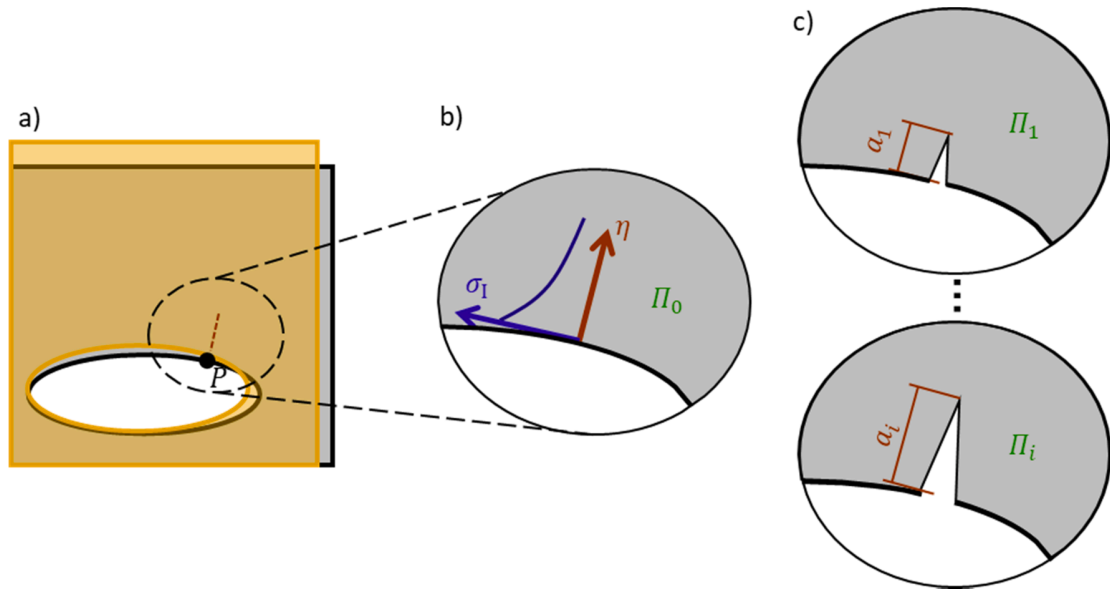
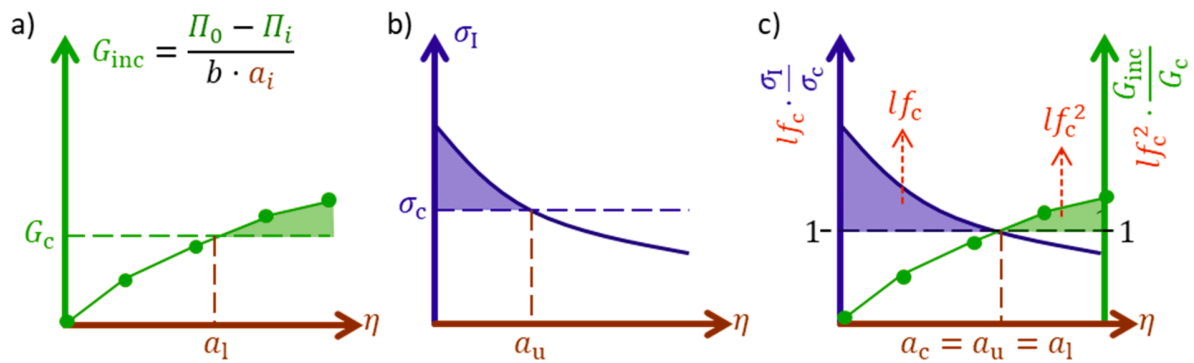


Fig. 1. For a given notch defined by the spline  $\vec{\Gamma}(s)$  the max. principal stress  $\sigma_1(\eta)$  as well as the strain energies  $\Pi(a)$  along the virtual crack path  $\eta$  are evaluated at a position  $s = s_p$ . This can be done either by the Full FEA approach that uses several FEM analyses or by our novel SLMM approach that calculates the displacement field  $\vec{u}(x,y)$  with a voxel-model and the notch curvature  $c(s)$  with the spline  $\Gamma(s)$ . These parameters are passed to a precomputed crack model that is computed once in an image space for a given material and predicts  $\sigma_1(\eta)$  and  $\Pi(a)$ .



**Fig. 2.** a) CC is applied on the position P on the notch surface. The crack grows normal to the surface into the material. b) The maximum principal stress  $\sigma_I(\eta)$  is evaluated along the crack path  $\eta$ . c) The virtual crack is introduced with various lengths  $a_i$  and the strain energy  $\Pi_i$  is computed for each  $a_i$ .



**Fig. 3.** a) According to the energy criterion, crack initiation is possible for a length  $a_i$  and above, where the incremental energy release rate  $G_{inc}$  is greater than the fracture energy  $G_c$ . b) According to the stress-criterion, crack initiation is possible up to a length  $a_u$ , when the max. principal stress  $\sigma_I$  is greater than the strength  $\sigma_c$ . c) The load is scaled by a load factor  $lf$  that is proportional to  $\sigma_I \propto lf$  and  $G_{inc} \propto lf^2$ . According to CC, the critical load factor  $lf_c$  is chosen, such that both criteria are fulfilled at the same crack length  $a_c$ .

principal stress  $\sigma_I(\eta)$  is additionally computed. With the calculated  $\Pi_i$  and  $\sigma_I(\eta)$  values, the CC is applied as described in section 2.1.

We use quadratic 2D elements with reduced integration. At the model borders, the mesh size is  $l_{el,1} = 0.1 \cdot l$  where  $l$  is the length of the model. In the region around the virtual crack, a fine mesh size  $l_{el,2} = 0.01 \cdot a_n$  is used. The mesh parameters  $l_{el,full} = (l_{el,1}, l_{el,2})$  are validated in section 3.1.3. During the virtual crack growth, we only add new nodes at one crack side, but do not remesh the model. This reduces numerical errors.

### 2.2.2. Local model

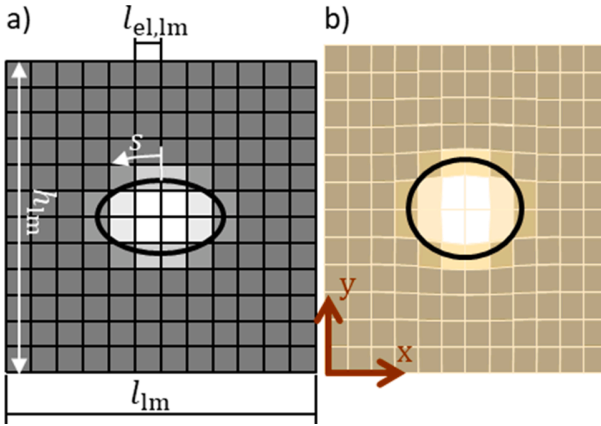
In contrast to the Full FEA approach, the novel SLMM approach uses two FEA models. The local model is one of them. We implemented it as a voxel model such as in Fig. 4a,b) because a pixel image of a notch can be easily transformed into a voxel mesh. Furthermore, a voxel mesh interpolates a displacement field  $\vec{u}(x,y)$  much faster than an unstructured mesh. Fig. 4a) depicts the material density  $\rho$ , which is almost zero  $\rho_{notch} = 0.01$  (white) inside the notch and one  $\rho_{solid} = 1$  (gray) in the full material region. The element Young's modulus  $E_{el} = \rho \cdot E$  weights the material Young's modulus  $E$  by the material density  $\rho$ . Fig. 4b) shows the deformed model under uniaxial vertical tension. The local model has a

length  $l_{lm}$  and a height  $h_{lm}$ . We chose the element size  $l_{el,lm} \approx 0.001 \cdot \sqrt{l_{lm} \cdot h_{lm}}$ , such that the model contains approximately  $n_{el,lm} \approx 10^6$  linear rectangular fully-integrated 2D elements. The mesh size is validated in section 3.1.3.

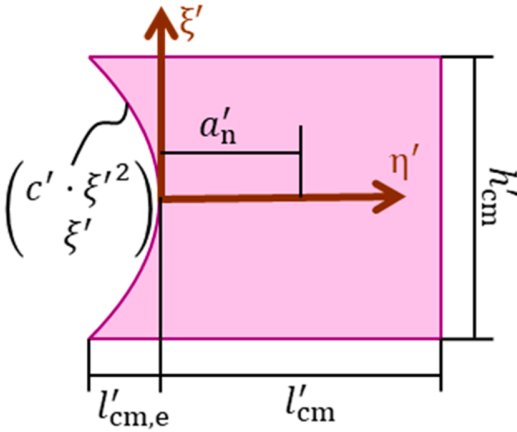
### 2.2.3. Crack model

The crack model is the second FEA model used by SLMM besides the local model. Fig. 5 shows the crack model and Table 1 lists its parameters. The crack model is defined in an  $\xi' \eta'$  image space. Quantities in the image space are marked by an apostrophe. Section 2.3 describes how the left curved side approximates a notch surface, how the Young's modulus  $E'$  and other dimensions are scaled to the real space, and how the curvature  $c'$  is used by a meta-model. The Poisson's ratio  $\nu$  depends on the material. We use a structured mesh with quadratic rectangular and reduced integrated 2D elements. The selection of the element size  $l'_{el,cm}$  as well as the number of introduced crack lengths  $n_{cracks}$  is argued in section 3.1.3.

The deformation modes shown in Fig. 6 define the boundary conditions of the crack model. A deformation mode.



**Fig. 4.** The local model is implemented as a voxel model. a) The undeformed mesh depicts the element density  $\rho$ , which is almost zero (white) inside the notch and one (gray) in the material. The element size  $l_{el,lm}$ , model length  $l_{lm}$ , and model height  $h_{lm}$  are shown. b) The displacement field  $\vec{u}(x, y)$  that results from a FEM analysis deforms the voxel mesh.



**Fig. 5.** Crack model with curvature  $c'$ , length  $l'_{cm}$ , height  $h'_{cm}$  and maximum crack length  $a'_n$ . The extension length  $l'_{cm,e}$  goes toward the curved surface. The crack model is defined in the  $\xi\eta'$ -image-space.

**Table 1**  
Crack model parameters.

Young's modulus	$E'$	1 MPa
Poisson's ratio	$\nu$	-1 to 0.5
Length	$l'_{cm}$	2 mm
Height	$h'_{cm}$	2 mm
Maximum crack length	$a'_n$	1 mm
Curvature	$c'$	-1 to 0.5/mm
Element size	$l'_{el,cm}$	0,01 mm
Number of cracks	$n_{cracks}$	10

$$\vec{u}'_i(\xi', \eta') = u'_{0,i} \begin{pmatrix} f_{\xi,i}(\xi', \eta') \\ f_{\eta,i}(\xi', \eta') \end{pmatrix} \quad (3)$$

defines the displacements in  $\xi'$ - and  $\eta'$ -direction with arbitrary 2D functions  $f_{\xi,i}(\xi', \eta')$  and  $f_{\eta,i}(\xi', \eta')$  respectively. These functions are scaled by a reference value  $u'_{0,i}$  that can be any nonzero value. The unit of  $\vec{u}'_i(\xi', \eta')$  is (mm) and the 2D functions  $f_{\xi,i}(\xi', \eta')$  and  $f_{\eta,i}(\xi', \eta')$  can have a unit we call "unit( $f_i$ )". The unit of  $u'_{0,i}$  is  $(\text{mm} \cdot \text{unit}(f_i))^{-1}$ .

To define the  $\xi'$ - and  $\eta'$ -displacements independently, one of the

functions  $f_{\xi,i}(\xi', \eta')$  or  $f_{\eta,i}(\xi', \eta')$  is set to zero for each deformation mode  $\vec{u}'_i(\xi', \eta')$ . Fig. 6 depicts the deformation modes  $\left\{ \vec{u}'_i(\xi', \eta') \mid i = 1, 2, 3, 6, 7, 8, 9, 10 \right\}$  for  $\xi$ -deformations on the left side. The  $\eta$ -function  $f_{\eta,i}(\xi, \eta)$  is set to zero for them. On the right side the deformation modes  $\left\{ \vec{u}'_i(\xi', \eta') \mid i = 4, 5, 11, 12, 13, 14, 15, 16 \right\}$  for  $\eta$ -deformations with  $f_{\xi,i}(\xi, \eta) = 0$  are shown.

The final deformation.

$$\vec{u}' = \sum_{\vec{u}'_i \in \mathbb{D}'} u'_{RP,i} \vec{u}'_i(\xi', \eta') \quad (4)$$

is a linear combination of deformation modes where dimensionless reference point displacements  $u'_{RP,i}$  act as scaling variables. Reference points are used in the FE software to apply a displacement as boundary condition or to evaluate reaction forces. This procedure, also known as the linear superposition principle, allows the results to be precomputed for each deformation mode independently and then simply be summed. The  $\vec{u}'_i(\xi', \eta')$  functions used in the analysis are collected in the set  $\mathbb{D}'$ . The final deformation  $\vec{u}'(u'_{RP,i}, \xi', \eta')$  defines the displacements along the boundaries of the crack model, except the left curved side which is not constrained.

The 2D functions are not limited to the  $\xi\eta'$  image space as  $f_i(\xi', \eta')$ , but are later used in the  $\xi\eta$  real space as  $f_i(\xi, \eta)$  as well. So, we omit the apostrophe for  $\xi$  and  $\eta$  when we speak of them. In this work, the 2D functions are generated using the scheme.

$$f(\xi, \eta) = \xi^{p_\xi} \cdot \eta^{p_\eta} \text{ with } p_\xi, p_\eta \in [0, 1, 2] \text{ and } p_\xi + p_\eta > 0. \quad (5)$$

Note that the units of the functions  $\text{unit}(f) = (\text{mm}^{p_\xi + p_\eta})$  depend on the powers  $p_\xi$  and  $p_\eta$  and consequently are not uniform.

For accurate predictions, the full set of deformation modes  $\mathbb{D}'_{all} = \left\{ \vec{u}'_i(\xi', \eta') \mid i = 1 \dots 16 \right\}$  is neither required nor recommended. By activating and deactivating each deformation mode, we found that the subset  $\mathbb{D}'_{imp} = \left\{ \vec{u}'_i(\xi', \eta') \mid i = 1 \dots 5 \right\}$  is capable to sufficiently fit the deformation field at the circular cavity shown in the results section 3.1.1 in Fig. 13 under uniaxial tension with a coefficient of determination of at least  $R^2 \geq 0.9$ . The effect of using  $\mathbb{D}'_{imp}$  instead of  $\mathbb{D}'_{all}$  is shown in section 3.1.3. Fig. 6 highlights  $\vec{u}'_i(\xi', \eta') \in \mathbb{D}'_{imp}$  by red frames. In the following section, we use only  $\mathbb{D}' = \mathbb{D}'_{imp}$ , and disregard other deformation modes.

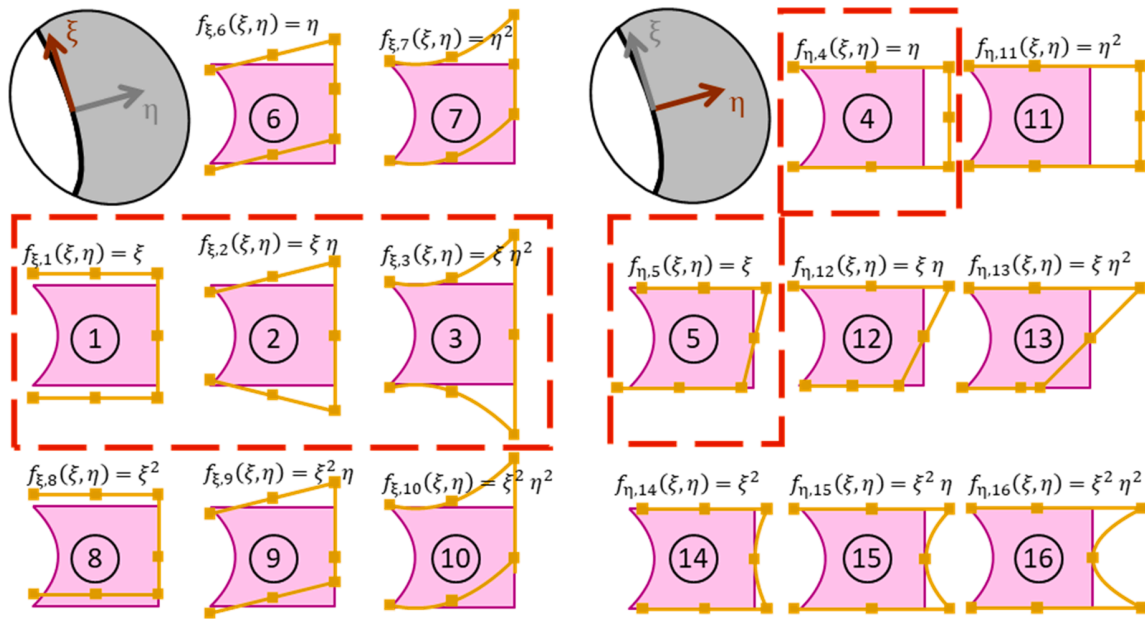
Knowing the deformation modes  $\mathbb{D}'$ , the crack model can be pre-computed. As shown in Fig. 1 the crack model contains a virtual crack with increasing length  $a'$ . Like for the Full FEA approach,  $1 + n_{cracks}$  virtual crack lengths  $a'_j$  are introduced from  $a'_0 = 0\text{mm}$  to  $a'_n$ . This procedure is repeated for each deformation mode. Therefore, the corresponding reference point displacements.

$$u'_{RP,ik} = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{else} \end{cases} \text{ with } i, k = 1 \dots |\mathbb{D}'| \quad (6)$$

of the  $k$ -th repetition are used in  $\vec{u}'(u'_{RP,ik}, \xi', \eta')$  to apply only the  $k$ -th deformation mode as a displacement-controlled boundary condition. The cardinality  $|\mathbb{D}'|$  is the number of deformation modes in the set  $\mathbb{D}'$ . The reference point reaction forces  $F'_{RP,ik}(a'_j)$  are computed and depend on the virtual crack length  $a'_j$ . The stiffness matrix.

$$K'_{ik}(a'_j) = F'_{RP,ik}(a'_j) \cdot \text{inv}(u'_{RP,ik}) \quad (7)$$





**Fig. 6.** Polynomial deformation modes for displacements in  $\xi$ -direction (left) and  $\eta$ -direction (right). The red framed deformation modes are important for the crack initiation and are collected in the set  $\mathbb{D}'_{\text{imp}}$ . All other modes are ignored in the following sections. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

is computed with  $F'_{\text{RP},ik}(a'_j)$  and the inverse  $\text{inv}(u'_{\text{RP},ik})$  of the reference point displacement matrix. For simplicity, we use the notation  $\vec{u}'_{\text{RP}} = (u'_{\text{RP},i})_i$ ,  $\vec{F}'_{\text{RP}}(a'_j) = (F'_{\text{RP},i}(a'_j))_i$  and  $\mathbf{K}'(a'_j) = (K'_{ik}(a'_j))_{i,k}$  with  $i, k = 1 \dots |\mathbb{D}'|$  and  $j = 0 \dots n_{\text{cracks}}$ . Knowing the stiffness matrix, the strain energy

$$\Pi'(a'_j, \vec{u}'_{\text{RP}}) = \vec{u}'_{\text{RP}}{}^T \mathbf{K}'(a'_j) \cdot \vec{u}'_{\text{RP}} \quad (8)$$

can be evaluated for any virtual crack length  $a'_j$  and arbitrary reference point displacements  $\vec{u}'_{\text{RP}}$ . The incremental energy release rate

$$\begin{aligned} G'_{\text{inc}}(0 \text{ mm}, \vec{u}'_{\text{RP}}) &:= \frac{0 \text{ mJ}}{\text{mm}^2} \\ &\quad \Psi'(a'=0 \text{ mm}) \\ G'_{\text{inc}}(a'_j, \vec{u}'_{\text{RP}}) &= \frac{\Pi'(a'_0, \vec{u}'_{\text{RP}}) - \Pi'(a'_j, \vec{u}'_{\text{RP}})}{b \cdot a'_j} \\ &= \frac{\vec{u}'_{\text{RP}}{}^T \mathbf{K}'(a'_0) - \mathbf{K}'(a'_j) \vec{u}'_{\text{RP}}}{b \cdot a'_j}, \forall a'_j > 0 \text{ mm} \end{aligned} \quad (9)$$

is computed by a matrix function  $\Psi'(a'_j)$ . The components of  $\Psi'(a'_j)$  are interpolated piecewise linearly along  $a'$ . The piecewise linear interpolation  $\Psi'(a')$  preserves the positive semi definiteness of  $\Psi'(a'_j)$ . This guarantees a non-negative incremental energy release rate  $G'_{\text{inc}}(a', \vec{u}'_{\text{RP}})$ .

In contrast to the *Full FEA* approach, the max. principal stress is not computed directly. Instead, the components of the 2D stress tensor

$$\sigma'_k(\eta') = \begin{bmatrix} \sigma'_{\xi\xi,k}(\eta') & \sigma'_{\xi\eta,k}(\eta') \\ \sigma'_{\eta\xi,k}(\eta') & \sigma'_{\eta\eta,k}(\eta') \end{bmatrix} \quad (10)$$

are interpolated piecewise linearly along the virtual crack path  $\eta'$  in the simulations with a virtual crack length of  $a'_0$ .

The final 2D stress tensor

$$\sigma'(\eta', \vec{u}'_{\text{RP}}) = \sum_{k=1 \dots |\mathbb{D}'|} \sigma'_k(\eta') \cdot u'_{\text{RP},i} \quad (11)$$

is a linear combination that weights the stress tensors  $\sigma'_k(\eta')$  with the corresponding reference point displacements  $u'_{\text{RP},i}$ . The max. principal stress  $\sigma'_1(\eta', \vec{u}'_{\text{RP}})$  is the highest eigenvalue of the stress tensor  $\sigma'(\eta', \vec{u}'_{\text{RP}})$ .

### 2.3. Scaling law Meta-Model (SLMM)

The scaling law meta-model uses the local model and the crack model described in the previous sections to predict the max. principal stress  $\sigma_1(\eta)$  and the incremental energy release rate  $G_{\text{inc}}(a)$ . The local model exists in the real space, whereas the crack model is defined in an image space. The next section describes how quantities are transformed from one space to another.

#### 2.3.1. Scaling approach

The dimensional analysis defines how a physical system is converted to another equivalent physical system [24]. In this work, we want to convert between the image space containing the crack model and the real space containing the local model. Therefore, scale factors  $\lambda_i$  are introduced for all dimensions. We use a dimension system with the two dimensions, length (mm) and stress (MPa), and introduce the two corresponding scale factors  $\lambda_l$  and  $\lambda_\sigma$ . Quantities and factors corresponding to the image space are marked with an apostrophe. To show how the scale factors are used, we first define a quantity in the image space.

$$Q' = q' \cdot \text{mm}^{p_1} \cdot \text{MPa}^{p_\sigma} \quad (12)$$

with a dimensionless factor  $q'$  and a unit ( $\text{mm}^{p_1} \cdot \text{MPa}^{p_\sigma}$ ) which is defined by its powers ( $p_1, p_\sigma$ ). To transfer the quantity to the real space.

$$Q = q \cdot \text{mm}^{p_1} \cdot \text{MPa}^{p_\sigma} = q' \cdot (\lambda_1 \cdot \text{mm})^{p_1} \cdot (\lambda_\sigma \cdot \text{MPa})^{p_\sigma}, \quad (13)$$

the length and stress units are multiplied by the scale factors. Such scale factors are commonly used to apply the CC [25]. It is crucial to convert all relevant quantities of a physical system because converted quantities must not be used alongside non-converted quantities.

The stress scale factor.

$$\lambda_\sigma(E) = \frac{E}{E'} \quad (14)$$

is the ratio between the material Young's modulus  $E$  and the Young's modulus  $E'$  used to simulate the crack model in the image space. The length scale factor.

$$\lambda_l(c', c) = \frac{c'}{c} \quad (15)$$

is chosen, such that the converted curvature of the crack model in the image space  $c' \cdot \lambda_l^{-1}$  fits the notch curvature in the real space  $c$ . If  $\lambda_l$  and  $c$  are known and  $c'$  is searched, it is possible to rearrange the above equation to.

$$c'(c, \lambda_l) = c \cdot \lambda_l \quad (16)$$

As shown in Fig. 7, crack models with various curvatures  $c'$  can be scaled by a factor  $\lambda_l$  such that they fit the curvatures  $c(s)$  at arbitrary positions  $s$  on the notch surface. In Fig. 7, the first crack model a) has a higher absolute curvature  $|c'| = 0.1 \text{mm}^{-1}$  than the second crack model b) with  $|c'| = 0.05 \text{mm}^{-1}$ . Therefore, the scale factor for model a) is twice as high as the one for model b) to fit the local curvature of the notch surface. At position  $s_1$ , both crack models are relatively small and since the virtual crack is introduced up to a length of half of the crack model  $a_n = l_{\text{cm}}/2$ , the max. principal stress  $\sigma_1(a)$ , and the incremental energy release rate  $G_{\text{inc}}(a)$  can be predicted only up to a small  $a_n$ . One possibility to reduce this limitation is to simulate several crack models with various curvatures  $c'$  and choose the best one, as described in section 2.3.5.

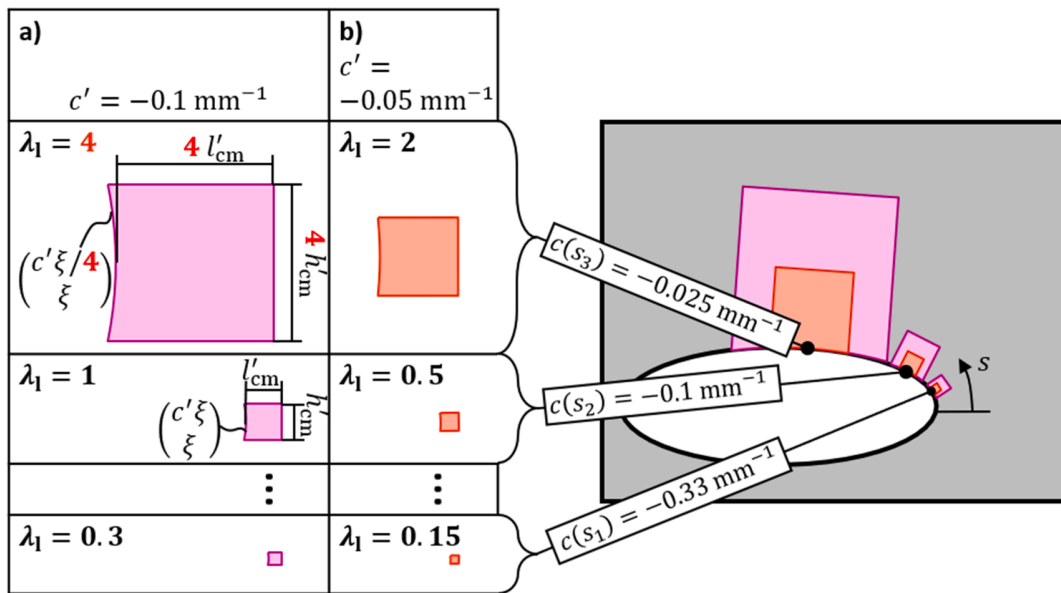


Fig. 7. The curvature in the crack model  $c'$  is set to  $-0.1 \text{mm}^{-1}$  and  $-0.05 \text{mm}^{-1}$  for two crack models depicted in the columns a) and b). The crack models can be scaled by a factor  $\lambda_l$  such that they fit the notch curvature  $c(s)$  at arbitrary positions  $s$  on the notch surface.

With the known scale factors  $\lambda_l$  and  $\lambda_\sigma$ , all other quantities can be transferred between the image and real space. The quantities maximum crack length  $a_n$ , cell length  $l_{\text{cm}}$ , and cell height  $h_{\text{cm}}$  all have the unit ( $\text{mm}^1 \cdot \text{MPa}^0$ ) and are scaled by  $\lambda_l = \lambda_l^1 \cdot \lambda_\sigma^0$  as.

$$\begin{aligned} a_n(\lambda_l) &= a'_n \cdot \lambda_l \\ l_{\text{cm}}(\lambda_l) &= l'_{\text{cm}} \cdot \lambda_l \\ h_{\text{cm}}(\lambda_l) &= h'_{\text{cm}} \cdot \lambda_l \end{aligned} \quad (17)$$

To scale a quantity in the opposite direction from the real space to the image space, the signs of the scale factor's powers are inverted. For example, the virtual crack length  $a$  and the coordinates  $\eta$  and  $\xi$  have the unit ( $\text{mm}^1 \cdot \text{MPa}^0$ ) and are scaled to the image space by  $\lambda_l^{-1}$ .

$$a'(a, \lambda_l) = a \cdot \lambda_l^{-1}$$

$$\eta'(\eta, \lambda_l) = \eta \cdot \lambda_l^{-1}$$

$$\xi'(\xi, \lambda_l) = \xi \cdot \lambda_l^{-1} \quad (18)$$

The deformation modes are converted first from the image space  $\vec{u}'_i(\xi', \eta')$ , to the real space  $\vec{u}_i(\xi, \eta)$ . Then, the corresponding dimensionless reference point displacements  $u_{\text{RP},i}$  are fitted in the real space.

To transfer the deformation modes  $\vec{u}'_i(\xi', \eta')$  from the image space to the real space  $\vec{u}_i(\xi, \eta)$ , the reference values  $u_{0,i}$  and the 2D functions  $f_i(\xi', \eta')$  must be scaled. The functions  $f_i(\xi', \eta')$  are simply converted to  $f_i(\xi, \eta)$  by inserting the real space coordinates  $\xi$  and  $\eta$ . The reference values  $u_{0,i}$  have units ( $\text{mm} \cdot \text{unit}(f_i)^{-1}$ ) that depend on the unit of  $f_i$ . Since the functions we use are polynomials  $f_i(\xi, \eta) = \xi^{p_{\xi,i}} \cdot \eta^{p_{\eta,i}}$ , their units can be expressed as ( $\text{mm}^{p_{\xi,i} + p_{\eta,i}}$ ). Consequently,  $u_{0,i}$  have units ( $\text{mm}^{1 - p_{\xi,i} - p_{\eta,i}}$ ) and are scaled by  $\lambda_l = \lambda_l^{1 - p_{\xi,i} - p_{\eta,i}} \cdot \lambda_\sigma^0$  as.

$$u_{0,i}(\lambda_l) = u'_{0,i} \cdot \lambda_l^{1 - p_{\xi,i} - p_{\eta,i}} \quad (19)$$

The transformed deformation modes.

$$\vec{u}'_i(\xi, \eta, \lambda_l) = u_{0,i}(\lambda_l) \cdot \begin{pmatrix} f_{\xi,i}(\xi, \eta) \\ f_{\eta,i}(\xi, \eta) \end{pmatrix} \in \mathbb{D} \quad (20)$$

and the dimensionless reference point displacements  $u'_{\text{RP},i}(u_{\text{RP},i}) = u_{\text{RP},i}$

are inserted into the final deformation in the real space.

$$\vec{u}(u_{RP,i}, \xi, \eta, \lambda_1) = \sum_{\vec{u}_j \in D} u_{RP,i} \vec{u}_i(\xi, \eta, \lambda_1). \quad (21)$$

The crack model provides the max. principal stress  $\sigma_1^i(a^i, \vec{u}_{RP}^i)$  and the incremental energy release rate  $G_{inc}^i(a^i, \vec{u}_{RP}^i)$  in the image space. Both quantities are transferred to the local model in the real space. The max. principal stress has the unit ( $\text{mm}^0 \cdot \text{MPa}^1$ ) and is scaled by  $\lambda_\sigma = \lambda_1^0 \cdot \lambda_\sigma^1$  to.

$$\sigma_1(\sigma_1^i, \lambda_\sigma) = \sigma_1^i \cdot \lambda_\sigma. \quad (22)$$

The incremental energy release rate has the unit ( $\text{mm}^1 \cdot \text{MPa}^1$ ) and is scaled by  $\lambda_1^1 \cdot \lambda_\sigma^1 = \lambda_1 \cdot \lambda_\sigma$ .

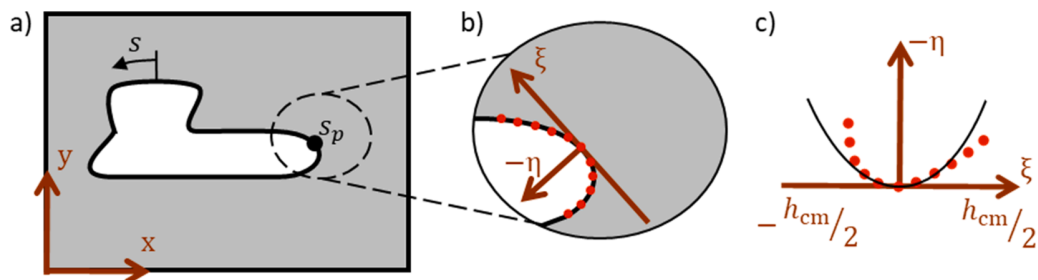
$$G_{inc}(G_{inc}^i, \lambda_1, \lambda_\sigma) = G_{inc}^i \cdot \lambda_1 \cdot \lambda_\sigma \quad (23)$$

### 2.3.2. Curvature fit of the local notch surface

In the previous section, we aligned scaled crack models to a notch surface, such that the scaled curvature of the crack model fits the curvature of the notch surface. Therefore, the curvature  $c$  of the notch surface is necessary. Fig. 8 shows how  $c$  is determined. The local model in Fig. 8a) contains a notch. A closed cubic B-spline  $\vec{\Gamma}(s) = (\Gamma_x(s), \Gamma_y(s))$  describes the surface of the notch. At a specific position  $s = s_p$ , the axis  $\xi$  tangential to the notch surface is computed as shown in Fig. 8b). The normal axis  $\eta$  is perpendicular to the tangential axis  $\xi$ . Next,  $n_{fit,c}$  points are sampled equidistant along the spline  $\vec{\Gamma}(s_i) = (x_i, y_i)$  in the interval  $s_i \in [s - \frac{h_{cm}}{2}, s + \frac{h_{cm}}{2}]$ . Our choice of  $n_{fit,c} = 11$  is argued in section 3.1.3. The height  $h_{cm}$  is chosen in an iterative process described in section 2.3.5. Fig. 8c) shows the sampled and transformed points  $(x_i, y_i) \rightsquigarrow (\xi_i, \eta_i)$  in the  $\xi\eta$ -coordinate system.

### 2.3.3. Displacement field fit

Once the curvature is extracted, the boundary conditions of the crack model are determined. To this end, the displacement field of the local model is computed by a FEM analysis. Fig. 9a) shows the resulting deformed local model. Fig. 9b) shows the scaled crack model placed at a specific position  $s_p$  on the notch. The displacements of the voxel mesh are interpolated onto  $n_{fit,u}$  points on the three straight edges of the crack model boundary. Our choice of  $n_{fit,u} = 28$  is argued in section 3.1.3. The interpolated displacements are then transformed into the  $\xi\eta$ -coordinate system, as shown in Fig. 9c). The final deformation  $\vec{u}(u_{RP,i}, \xi, \eta, \lambda_1)$  and an offset vector  $\vec{u}_{off} = (\xi_{off}, \eta_{off})$  fit the transformed displacements  $\vec{u}_j$  using the method of least squares by varying the reference point displacements  $\vec{u}_{RP}$  and  $\vec{u}_{off}$ .



**Fig. 8.** Computation of the local notch curvature  $c(s)$ . a) A position on the notch surface is specified by  $s = s_p$ . b) A coordinate system with a tangential axis  $\xi$ , and a normal axis  $-\eta$  is defined, and  $n_{fit,c}$  points are sampled on the notch surface. c) The sampled points are transformed to the  $\xi\eta$ -coordinate system and are fitted by a quadratic polynomial using the method of least squares.

$$\min_{\vec{u}_{RP}, \vec{u}_{off}} \sum_j^{n_{fit,u}} \|\vec{u}_j - \vec{u}(u_{RP,i}, \xi, \eta, \lambda_1) - \vec{u}_{off}\|_2. \quad (24)$$

The offset vector  $\vec{u}_{off}$  handles the translation which does not introduce any deformation to the crack model. The fitted reference point displacements  $\vec{u}_{RP}(s, h_{cm}, l_{cm}, c)$  depend on the position  $s$  as well as on the cell height  $h_{cm}$ , the cell length  $l_{cm}$ , and curvature  $c$ .

### 2.3.4. Meta-Model

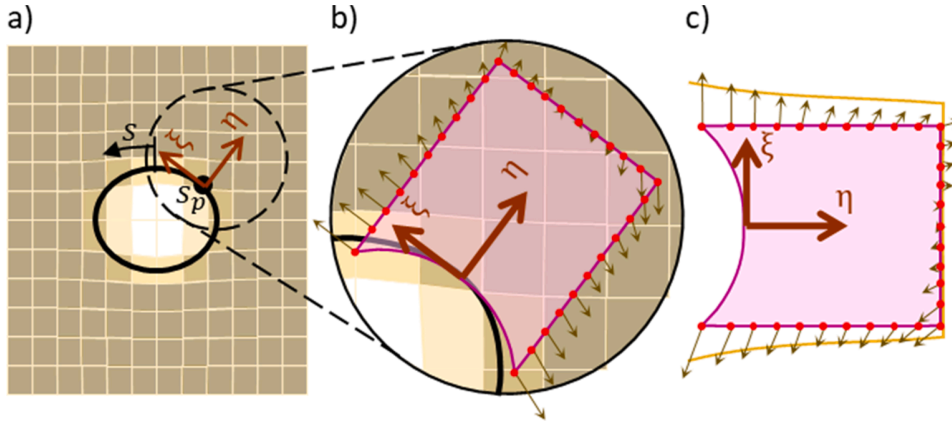
As demonstrated in Fig. 7, it is possible to scale and align a crack model to any notch regardless of the curvature  $c$  as long as both are either convex or concave. However, for notches with a high absolute curvature  $|c|$  and crack models with a small absolute curvature  $|c'|$ , the scaled crack model can only predict a crack up to a small length  $a_n$ . Therefore, we recommend using crack models with various curvatures  $c'$  and then choosing the best one. As shown in Fig. 10, the proposed meta-model consists of  $n_{curv}$  crack models with curvatures  $c'_i$ , ranging from  $c'_{min} = -1\text{mm}^{-1}$  to  $c'_{max} = 0.5\text{mm}^{-1}$ . We set  $n_{curv} = 16$ , such that crack models are computed in  $0.1\text{mm}^{-1}$  steps. The minimum curvature  $c'_{min}$  is determined by the sharp corners of the curved side that distorts the finite elements, whereas the displacement fit only works properly if  $c'_{max}$  is not too big.

Each crack model is computed as described in section 2.2.3. The components of the stiffness matrix  $K^i(a^i, c^i)$ , as well as the components of the stress tensor  $\sigma^i(\eta^i, \vec{u}_{RP}^i, c^i)$  and incremental energy release rate  $G_{inc}^i(a^i, \vec{u}_{RP}^i, c^i)$  are computed for each model and are interpolated piecewise linearly for the curvature  $c^i$ . The max. principal stress  $\sigma_1^i(\eta^i, \vec{u}_{RP}^i, c^i)$  is the greatest eigenvalue of  $\sigma^i(\eta^i, \vec{u}_{RP}^i, c^i)$ .

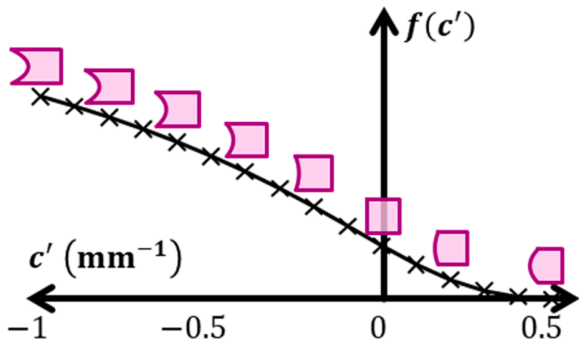
### 2.3.5. Application of the SLMM

In the previous section, crack models with various curvatures are precomputed in advance and an interpolation scheme is given. In this section, we describe an iterative process to find the best crack model in the meta-model for one position on a notch. The best crack model predicts  $G_{inc}(a)$  and  $\sigma_1(\eta)$  up to a possibly large virtual crack length  $a_n$  and fulfills all constraints defined in the next paragraph.

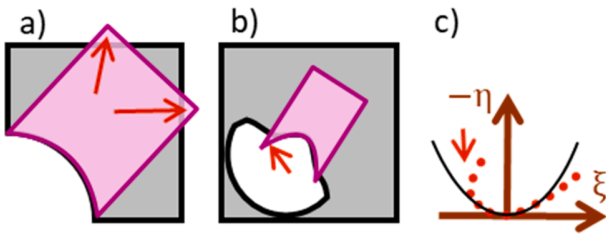
The code fragment (27) contains pseudo code to find the best crack model. First, the height  $h_{cm}$  of the scaled crack model is set to a multiple  $\varphi_{min}$  of the voxel element size  $l_{el,lm}$ , because we observed that the scaled crack model must cover a minimum number of voxel elements to calculate the reference point displacements properly. Our choice of  $\varphi_{min} = 10$  is argued in section 3.1.3. Next, the length scale factor  $\lambda_1$  is computed. With  $\lambda_1$ , all other quantities such as the cell length  $l_{cm}$ , the curvature  $c$ , and the reference point displacements  $\vec{u}_{RP}$  can be calculated in the real space as well as in the image space as  $l_{cm}^i, c^i$ , and  $\vec{u}_{RP}^i$ .



**Fig. 9.** Computation of the reference point displacements  $\vec{u}_{RP}$ . a) Displacement field  $\vec{u}(x, y)$  of the voxel mesh. At the position  $s = s_p$ , a coordinate system  $\xi\eta$  is aligned. b) The crack cell is placed onto the notch surface and the displacements of  $\vec{u}(x, y)$  are interpolated at  $n_{fit,u}$  sampling points  $(\xi_j, \eta_j)$  on the crack model border. c) The interpolated displacements  $\vec{u}_j$  (arrows) are fitted by a linear combination of the deformation modes (golden line) defined in Fig. 6.



**Fig. 10.** Crack models with various curvatures  $c'$  are evaluated. Results  $f(c')$ , like  $G'_{inc}$ , and  $\sigma'$  are interpolated piecewise linearly.



**Fig. 11.** Invalid crack cell alignments. a) The scaled crack cell exceeds the local model. b) The scaled crack cell intrudes into the notch. c) Undercut distorts the curvature fit.

Then, we can check if the scaled crack model violates one of the following constraints:

1. The scaled crack model must not exceed the local model as shown in Fig. 11a)
2. The scaled crack model must not intrude into the notch too much as shown in Fig. 11b). We allow at most two sampling points of the displacement fit to lie inside the notch.
3. The notch must not have an undercut in the region where the curvature fit is performed as shown in Fig. 11c). Consequently, the  $\xi_i$  values of the sampled points must either increase or decrease monotonically with the sampled  $s_i$  values.
4. The curvature in the image space must fulfill  $c'_{min} \leq c' \leq c'_{max}$ .

5. The curvature fit  $\eta(\xi, c) = c \cdot \xi^2$  should approximate the notch surface sufficiently. To check this, the variance of the fitted curvature

$$\text{Var}(c) = \frac{\sum_i^{n_{fit,c}} (\eta_i - c \cdot \xi_i^2)^2}{(n_{fit,c} - 1) \cdot \sum_i^{n_{fit,c}} \xi_i^4} \quad (25)$$

is computed for the sampled  $\xi_i$  and  $\eta_i$  values [26]. An accurate fit of the notch surface fulfills.

$$\sqrt{\text{Var}(c)} \cdot h \leq \Delta_{abs} + \Delta_{rel} \cdot |c \cdot h|. \quad (26)$$

In this work we use conservative settings for the absolute tolerance of  $\Delta_{abs} = 0.01$  and for the relative tolerance of  $\Delta_{rel} = 0.1$ .

If one of the constraints 1, 3, 4 or 5 is violated the iteration is stopped. If constraint 2 is not violated, the current length scale factor is assigned to  $\lambda_{l,best} := \lambda_l$ . Then, we try a larger scaled crack cell  $h_{cm} := h_{cm} \cdot (1 + \Delta_{inc})$  and repeat the loop. Our choice of  $\Delta_{inc} = 0.1$  increases the scaled crack cell in each iteration by 10%. After the iteration stopped,  $\lambda_{l,best}$  is used as length scale factor.

$$h_{cm} := \varphi_{min} \cdot l_{el,lm}$$

repeat loop:

$$\lambda_l := \frac{h_{cm}}{h'_{cm}}$$

If constraint 1 or 3 or 4 or 5 is violated:  
exit repeat loop

If constraint 2 is not violated:

$$\lambda_{l,best} := \lambda_l$$

$$h_{cm} := h_{cm} \cdot (1 + \Delta_{inc})$$

$$\lambda_l := \lambda_{l,best}$$

$$h_{cm} := \lambda_l \cdot h'_{cm}$$

(27)

Now, all quantities of the crack model in the real space and in the image space are calculated such as the curvatures  $c(s, h_{cm})$ ,  $c'(c, \lambda_l)$  and the reference point displacements  $\vec{u}_{RP}(s, h, l, c)$ ,  $\vec{u}'_{RP}(\vec{u}_{RP})$ . The crack model predicts the max. principal stress  $\sigma'_1(\eta', \vec{u}'_{RP}, c')$  along the virtual

crack path  $\eta^i(\eta, \lambda_1)$  and the incremental energy release rate  $G_{inc}^i(a', \vec{u}_{RP}^i, c^i)$  for various virtual crack lengths  $a^i(a, \lambda_1)$ . After those values are transformed back to the real space to  $G_{inc}(G_{inc}^i, \lambda_1, \lambda_\sigma)$  and  $\sigma_1(\sigma_1^i, \lambda_\sigma)$ , the CC computes a critical crack length  $a_c$  and a failure index  $FI$ .

#### 2.4. Boundary relaxation

When a crack grows, the displacement field in the whole component is influenced. However, our *displacement-controlled* (DC) submodel-like approach SLMM only considers the region inside the scaled crack model near the crack and neglects the additional deformation outside. So far, we assumed that the influence of the crack growth on the global displacement field outside the scaled crack model is neglectable. This assumption is only fulfilled if the crack is considerably smaller than the crack model. However, by introducing larger virtual cracks that can have a length up to half of the crack model length, this assumption is violated and the DC approach underestimates  $G_{inc}(a)$ , as can be seen in Fig. 12.

Fig. 12 shows two scaled crack models of different sizes as gold-colored curves (DC) used for the same position on a notch. The two scaled crack models are expected to return the same  $G_{inc}(a)$ . The first model predicts  $G_{inc}(a)$  up to a virtual crack length of  $a_{n,1}$ , whereas the larger second model predicts  $G_{inc}(a)$  up to  $a_{n,2}$ . Since the first model is smaller than the second model, the first model underestimates  $G_{inc}(a)$  for the same crack length  $a$  more than the second model, because it considers the influence of the crack growth in a smaller region inside the model and neglects it in a greater region outside the model. The deviation between the smaller and the larger model is most visible at the maximum virtual crack  $a_{n,1}$  length of the first model.

##### 2.4.1. Force-controlled boundary conditions

An alternative idea to the DC approach would be a *force-controlled* (FC) approach, where reference point forces  $\vec{F}_{RP}$  are applied instead of fixing the reference point displacements  $\vec{u}_{RP}(s, h, l, c)$ . The easiest way to do so is to compute the reference point displacements  $\vec{u}_{RP}(\vec{u}_{RP}) = \vec{u}_{RP}$  in the image space first because the stiffness matrix  $K^i(a, c^i)$  in the image space is already known. For the unnotched model with a virtual crack length  $a^i = 0$ mm, the boundary conditions are not affected by a crack, so the DC and FC approaches are identical. The unnotched model yields the reference point forces  $\vec{F}_{RP,FC}^i(K_0^i, \vec{u}_{RP}^i) = K_0^i \cdot \vec{u}_{RP}^i$  with  $K_0^i(c^i) = K^i(a^i = 0\text{mm}, c^i)$  and the displacement-controlled  $\vec{u}_{RP}^i$ . The FC

forces  $\vec{F}_{RP,FC}^i(K_0^i, \vec{u}_{RP}^i)$  and the stiffness matrix  $K^i(a^i, c^i)$  result in the FC reference point displacements.

$$\vec{u}_{RP,FC}^i(K^i, \vec{F}_{RP,FC}^i) = K^{i-1} \cdot \vec{F}_{RP,FC}^i. \quad (28)$$

Since  $K^i(a^i, c^i)$  and thus  $\vec{u}_{RP,FC}^i(K^i, \vec{F}_{RP,FC}^i)$  depend on the virtual crack length  $a^i$ , the incremental energy release rate  $G_{inc}^i(a^i, \vec{u}_{RP,FC}^i, c^i)$  can be computed along the virtual crack length  $a^i(a, \lambda_1)$ .

In contrast, the max. principal stress  $\sigma_1^i(\eta^i, \vec{u}_{RP,FC}^i(K_0^i, \vec{F}_{RP,FC}^i))$  is evaluated for the unnotched model with a virtual crack length  $a^i = 0$ mm and a stiffness matrix  $K_0^i(c^i)$ .  $G_{inc}^i(a^i, \vec{u}_{RP,FC}^i, c^i)$  and  $\sigma_1^i(a^i, \vec{u}_{RP}^i)$  are then scaled to the local model to yield  $G_{inc}(G_{inc}^i, \lambda_1, \lambda_\sigma)$  and  $\sigma_1(\sigma_1^i, \lambda_\sigma)$ . As depicted in Fig. 12 as dark blue curves, the FC approach overestimates  $G_{inc}$ . While  $\sigma_1$  is independent of the DC or FC approach,  $G_{inc}$  is underestimated by DC and overestimated by FC.

##### 2.4.2. Scaling law Meta-Model with auto-controlled boundary conditions (SLMM + AC)

The *auto-controlled* (AC) approach mixes the DC and FC approaches. Therefore, mixed reference point displacements  $\vec{u}_{RP,MC}^i(\vec{u}_{RP}^i, \vec{u}_{RP,FC}^i, w) = (1-w) \cdot \vec{u}_{RP}^i + w \cdot \vec{u}_{RP,FC}^i$  weight the DC and FC reference point displacements  $\vec{u}_{RP}^i(\vec{u}_{RP}, \lambda_1)$  and  $\vec{u}_{RP,FC}^i(K^i, \vec{F}_{RP,FC}^i)$  by a weighting factor  $w$ .

As can be seen in Fig. 12, the incremental energy release rates of two scaled crack models (1) and (2) diverge the most at a virtual crack length  $a_{n,1}$  for the golden DC and the dark blue FC approach. The AC approach tries to minimize divergence between (1) and (2) by varying the weighting factor  $w$  in an interval  $w \in [0, 1]$ . Therefore, we continuously insert previously defined quantities  $(a^i, \lambda_\sigma, G_{inc}^i, \dots)$  into the incremental energy release rate.

$$G_{inc}(G_{inc}^i, \lambda_1, \lambda_\sigma) \rightarrow G_{inc}\left(G_{inc}^i(a^i(\dots), \vec{u}_{RP,MC}^i(\dots), c^i(\dots)), \lambda_1, \lambda_\sigma(\dots)\right), \quad (29)$$

such that only the independent quantities virtual crack length  $a$ , position on the notch  $s$ , material's Young's modulus  $E$ , length scale factor  $\lambda_1$  and weight factor  $w$  remain free and we can express  $G_{inc}(a, s, E, \lambda_1, w)$  as a function of these independent quantities. The size of the two scaled crack models is controlled by the length scale factor  $\lambda_1$ . The first model is half as big as the second model, so the length scale factor is  $\lambda_{1,1} = 0.5 \cdot \lambda_{1,2}$ . The length scale factor  $\lambda_{1,2}$  of the second model is computed as described in section 2.3.

We can now compute two incremental energy release rates  $G_{inc}(a, s, E, \lambda_{1,1}, w)$  for the smaller-scaled crack model (1) in Fig. 12 and  $G_{inc}(a, s, E, \lambda_{1,2}, w)$  for the bigger scaled crack model (2) up to a max. crack length of  $a_{n,1} = a_n(\lambda_{1,1})$ . As already mentioned, the incremental energy release rates of the two models of different sizes deviate from each other for the DC approach with  $w = 0$ . Furthermore, the incremental energy release rates deviate also for the FC approach with  $w = 1$ . However, there is one weighting factor  $w$  at which the incremental energy release rates fit together. The weighting factor  $w$  can be found by minimizing the deviation of the incremental energy release rates at a virtual crack length  $a_{n,1}$

$$\min_w \|G_{inc}(a_{n,1}, s, E, \lambda_{1,1}, w) - G_{inc}(a_{n,1}, s, E, \lambda_{1,2}, w)\|_2. \quad (30)$$

As shown in Fig. 12, these red SLMM + AC curves do not deviate so

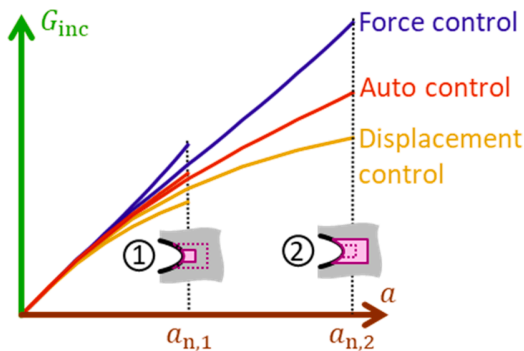
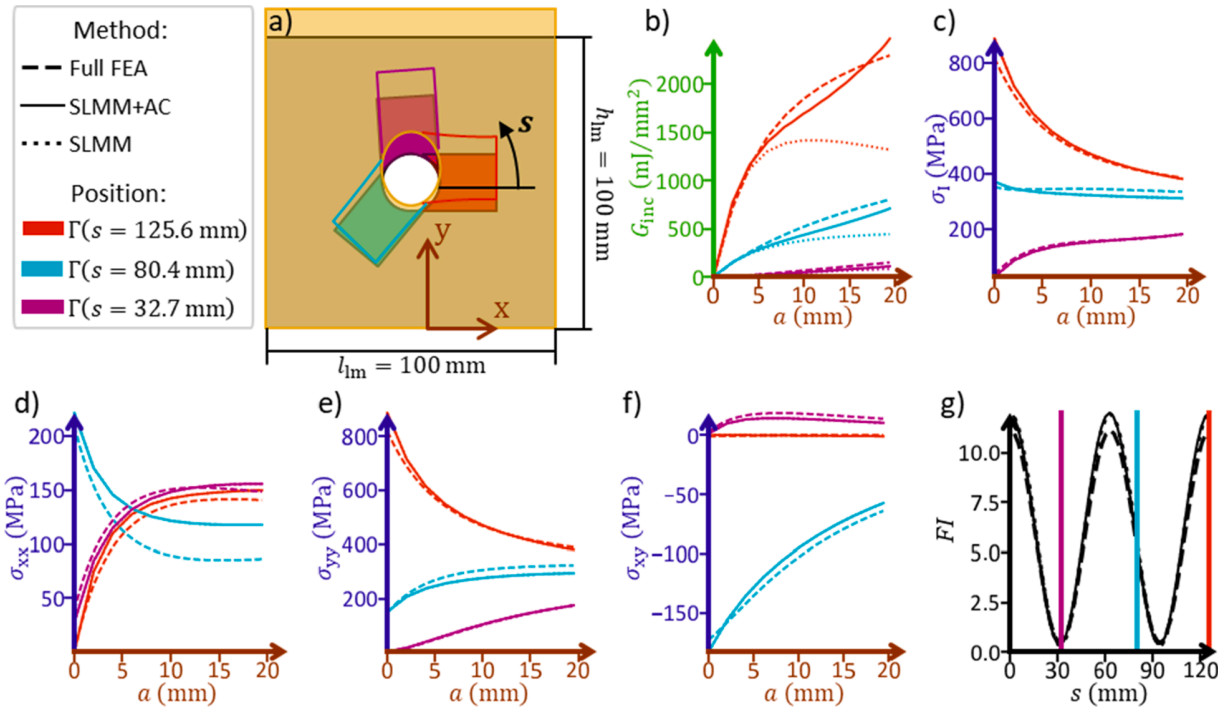


Fig. 12. Two crack models (1) and (2) of different sizes are applied to the same location on a notch. Both crack models predict the incremental energy release rate  $G_{inc}(a)$  both force-controlled (FC) and displacement-controlled (DC). At  $a_{n,1}$ , both FC and DC are inconsistent. The Auto-Controlled (AC) approach tries to minimize this inconsistency at  $a_{n,1}$  and therefore gains more accurate results.



**Fig. 13.** Application of the Full FEA, the auto-controlled SLMM (SLMM + AC) and the displacement-controlled SLMM (SLMM) at a circular hole. A strain of  $\varepsilon_{xx} = 0$ ,  $\varepsilon_{yy} = 0.1$  and  $\gamma_{xy} = 0$  is applied to the model borders. SLMM underestimates  $G_{inc}$  at longer virtual crack lengths in b), whereas the auto-controlled boundary conditions of SLMM + AC are a good approximation. The max. principal stress  $\sigma_1$  in c) is computed out of the stress components in d), e) and f) and shows good agreement of the SLMM/SLMM + AC approaches compared to the Full FEA.

much from each other, and they are a better approximation for  $G_{inc}$ . The weight factor is computed for each position on the notch separately and is, therefore, a function  $w(s)$  of the notch position  $s$ . In the further analysis, the results obtained from the scaled crack model (2) are used, because they cover a wider virtual crack length range from 0mm to  $a_{n,2}$ .

### 3. Results

The aim of this work is to provide a novel method to apply the CC in an efficient, versatile, and accurate manner. In this section, we first compare the accuracy of the novel SLMM approach with a Full FEA approach and experimental data. Then, we consider the efficiency and finally we explain when SLMM can be used for different materials given in Table 2. If not stated otherwise, we use a plane stress state and a Poisson's ratio of  $\nu = 0.35$ .

#### 3.1. Accuracy of SLMM and SLMM + AC

To assess the accuracy of SLMM and SLMM + AC, we compare their results to those of the Full FEA approach for two notch geometries. A circular hole geometry is validated with experimental data out from the literature. Furthermore, model parameters are validated with the help of

**Table 2**

Mechanical material properties. Literature provides values for  $K_{Ic}$  that are converted to the fracture energy by  $G_c = K_{Ic}^2/E$  under the assumption of a plane stress state. Irwin's length  $l_{ch} = K_{Ic}^2/\sigma_c^2$  refers the toughness  $K_{Ic}$  to the strength  $\sigma_c$ .

	$G_c \left( \frac{J}{m^2} \right)$	$\sigma_c$ (MPa)	$E$ (MPa)	$l_{ch}$ (mm)
PMMA [15]	1298	70.5	2,960	0.773
PS [15]	632.3	30	3,100	2.18
Al <sub>2</sub> O <sub>3</sub> [27,28]	40	400	300,000	0.75

a complex-shaped notch.

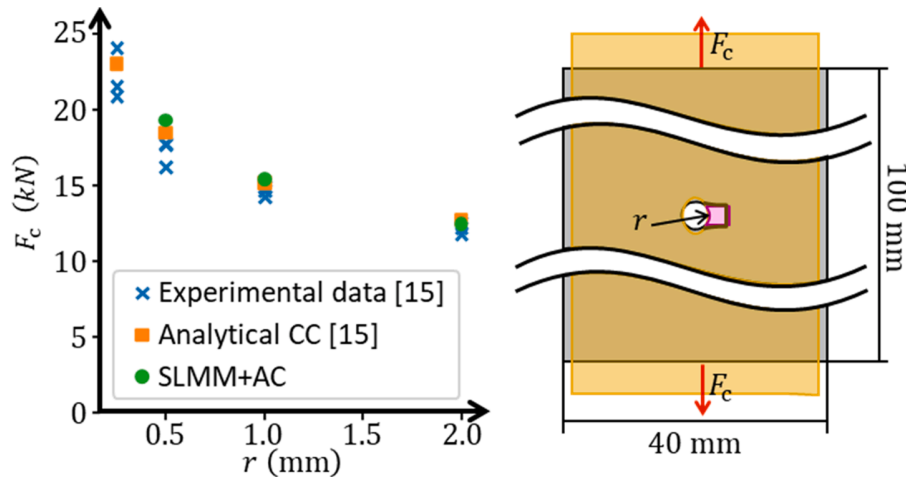
#### 3.1.1. Circular notch

Fig. 13a) shows a rectangular model with 100mm side length that contains a circular hole with a radius of  $r = 20$ mm. The boundary conditions at the model margins are defined by the strains  $\varepsilon_{xx} = 0$ ,  $\varepsilon_{yy} = 0.1$  and  $\gamma_{xy} = 0$ . The plate material is PMMA as defined in Table 2. The stresses and the incremental energy release rate are predicted by Full FEA, SLMM, and SLMM with auto-controlled boundary conditions (SLMM + AC) at three positions on the cavity. In addition, all three approaches calculate the failure index  $FI$  over the entire circular hole. The height of the scaled crack cells is set to  $h = 38.7$ mm for SLMM and SLMM + AC.

Fig. 13 shows good agreement between Full FEA, SLMM + AC, and SLMM. The incremental energy release rate  $G_{inc}$  of SLMM in Fig. 13b) shows a big deviation compared to Full FEA at longer virtual crack lengths  $a$ , whereas SLMM + AC agrees better with Full FEA due to the boundary relaxation approach. This approach influences  $G_{inc}$  but neither the stress components  $\sigma_{xx}$ ,  $\sigma_{yy}$ , and  $\sigma_{xy}$  in Fig. 13e-f) nor  $\sigma_1$ . Consequently, the stresses are identical for SLMM and SLMM + AC. At the cyan position in Fig. 13d), the stress component  $\sigma_{xx}$  of SLMM and SLMM + AC differ from the Full FEA values. However, this influences the max. principal stress  $\sigma_1$  in Fig. 13c) only by  $-7.7\%$ . The most critical failure index  $FI$  occurs at the red position and is overestimated by SLMM and SLMM + AC by  $7.96\%$  and  $7.98\%$ , respectively.

SLMM + AC agrees well with full FEA computations of a circular hole. We further validate SLMM + AC using the work of Sapora et al. [15], who provide experimental data as well as analytical CC predictions for specimens shown in Fig. 14. The specimens are made of PMMA and contain a circular hole of various radii  $r$ . A crack initiates at the hole when a critical force  $F_c$  is applied.

Since the specimens have a thickness of 10mm, we assume plane strain conditions and compute the fracture energy as  $G_c = K_{Ic}^2(1-\nu^2)/E$ .



**Fig. 14.** The right image shows a specimen with a thickness of 10 mm and a circular hole of a certain radius  $r$ . Crack initiation occurs at the circular hole once a critical force  $F_c$  is applied. Sapora et al. [15] provide experimental data and analytical CC predictions for various radii  $r \in [0.25, 0.5, 1.0, 2.0]$  mm. SLMM + AC agrees well with values provided in the literature, except for  $r = 0.25$  mm at which SLMM + AC is not applicable, because of a restriction discussed in section 3.3.

The Poisson's ratio is  $\nu = 0.38$ . We used Abaqus [29] to simulate the displacement field in the local model and then applied SLMM + AC to the right side of the circular hole.

Good agreement is obtained between SLMM + AC predictions, analytical CC predictions, and experimental data. The maximum deviation between SLMM + AC and analytical CC is 4.5% for a hole radius  $r = 0.5$  mm. However, SLMM + AC is not applicable for the smallest radius  $r = 0.25$  mm, because of a restriction discussed in section 3.3.

### 3.1.2. Complex-shaped notch

The following Fig. 15a-c) illustrates the SLMM + AC and Full FEA results of a complex-shaped notch under three load cases. In the first row, a strain of  $\epsilon_{xx} = 0.1, \epsilon_{yy} = 0.0, \gamma_{xy} = 0.0$  is applied to the model boundaries, whereas in the second row,  $\epsilon_{xx} = 0.0, \epsilon_{yy} = 0.1, \gamma_{xy} = 0.0$  is used and the third row shows the deformation, due to a strain of  $\epsilon_{xx} = 0.0, \epsilon_{yy} = 0.0, \gamma_{xy} = 0.1$ . The plate material is PMMA as defined in Table 2 for all three cases. The first picture in each row depicts the deformed notch and defines and enumerates critical positions. The second plot in each row shows the failure index  $FI$  over the position of the Full FEA and the SLMM + AC approaches. At positions close to  $s = 120$  mm, the geometry has a complex detail highlighted by point A in Fig. 15a) that leads to a discrepancy between both methods. The discrepancy is a result of the approximations made during the deformation and curvature fit. However, SLMM + AC finds all critical positions. The right plots show the computation of CC for the notch position with the highest load factor. Whereas the stresses of the SLMM + AC fit the stresses of the Full FEA well, the incremental energy release rates deviate for longer virtual cracks. The auto-controlled boundary conditions cannot fully compensate for the assumption made by the submodel-like approach, and this leads to a deviation.

### 3.1.3. Validation of model parameters

At many points during the development of the SLMM method, it was necessary to make assumptions. They were tested as the tool was developed. Here, we justify the selection of the parameter values varied. Those parameters are the number of sampling points in the curvature fit  $n_{fit,c}$ , the number of sampling points in the displacement fit  $n_{fit,u}$ , the number of introduced crack lengths  $n_{cracks}$ , the set containing the deformation modes  $\mathbb{D}$ , the minimum size ratio of the scaled crack model  $\varphi_{min}$ , the mesh size in the crack model  $l_{el,cm}$ , the number of elements in the local model  $n_{el,lm}$  and the mesh size of the reference model  $l_{el,full}$ . Their influence on the max. principal stress  $\sigma_1$  and on the incremental energy release rate  $G_{inc}$  is listed in Table 3.

Therefore, one parameter  $\nu$  at a time is set once to its default value  $\nu = \nu_1$  and once to a modified value  $\nu = \nu_2$ . SLMM and Full FEA analyses are performed with both settings  $\nu_1$  and  $\nu_2$  in the model shown in Fig. 15a) at position (1) and the max. principal stress  $\sigma_1(\eta, \nu)$  as well as the incremental energy release rate  $G_{inc}(a, \nu)$  are calculated. The maximum relative effect on  $\sigma_1(\eta, \nu)$  and  $G_{inc}(a, \nu)$  over the virtual crack path  $\eta$  or the virtual crack length  $a$  are computed as.

$$\begin{aligned} \Delta\sigma_1 &= 100\% \cdot \max_{\eta} \{1 - \sigma_1(\eta, \nu_1) / \sigma_1(\eta, \nu_2)\} \\ \Delta G_{inc} &= 100\% \cdot \max_a \{1 - G_{inc}(a, \nu_1) / G_{inc}(a, \nu_2)\} \end{aligned} \quad (31)$$

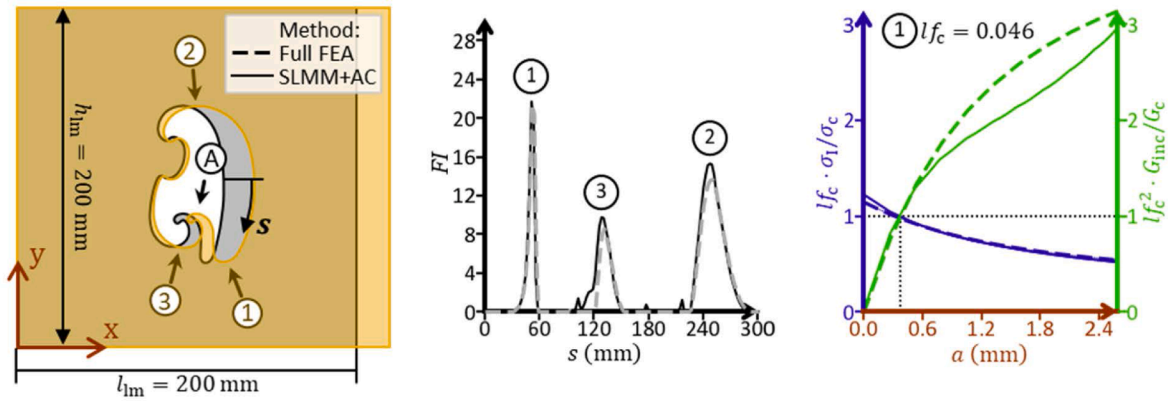
The observed effect is less than 1% and hence neglectable  $n_{fit,c}$ ,  $n_{fit,u}$ ,  $l_{el,cm}$  and  $l_{el,full}$ . The results of the five deformation modes in  $\mathbb{D}_{imp}$  differ at most 2.5% from the results of the 16 deformation modes in  $\mathbb{D}_{all}$ . The number of voxel elements  $n_{el,lm}$  in the local model leads to an acceptable effect of at most 3.9%.

The number of virtual crack lengths  $n_{cracks}$  introduced in the crack model does not influence  $\Delta\sigma_1$ , but significantly influences  $\Delta G_{inc}$ . The piecewise linear interpolation  $G_{inc}(a, \nu_2)$  does not look smooth, which explains the high effect. However, the interpolation  $G_{inc}(a, \nu_1)$  looks smooth.

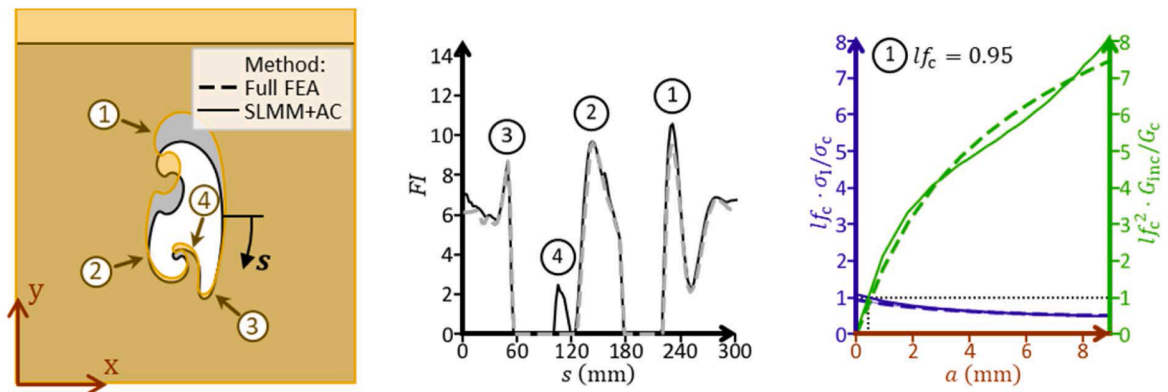
As described in section 2.3.5, the scaled crack model must be at least  $\varphi_{min} = 10$  times larger than the voxel element size, because each scaled crack model must cover a minimum number of voxel elements to fit the reference point displacements  $\vec{u}_{RP}$  properly. At position (1), the scaled crack model exactly fulfills this minimum requirement of being  $\varphi_{min}$  times larger than a voxel element. Consequently,  $\vec{u}_{RP}$  is fitted with  $\varphi_{min} = \nu_1$  not as accurately as with  $\varphi_{min} = \nu_2$ , which leads to a larger scaled crack model that covers more voxel elements. However, for  $\varphi_{min} = \nu_2$  the notch curvature  $c$  is fitted not as accurately as with  $\varphi_{min} = \nu_1$ . This dilemma of either fitting  $\vec{u}_{RP}$  or  $c$  accurately, is not completely resolved at position (1) and leads to a high relative error that is also the main cause for the deviation of  $G_{inc}$  in Fig. 15a).

## 3.2. Efficiency of SLMM + AC

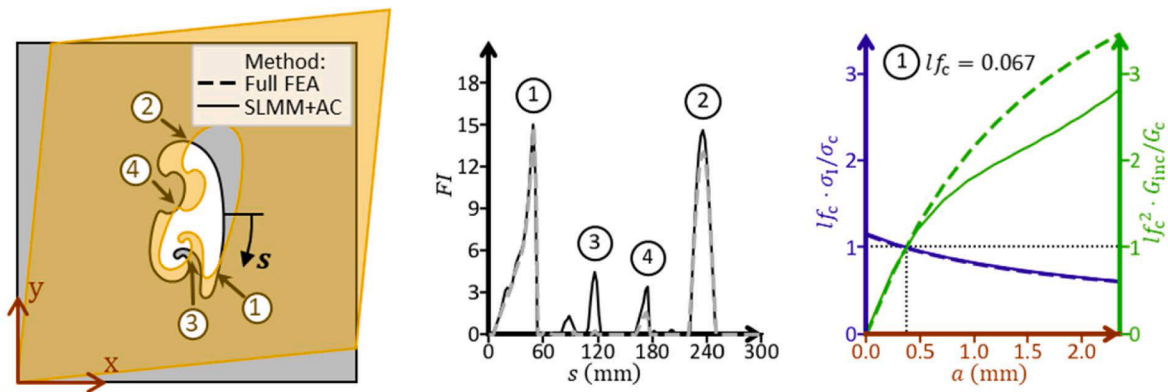
A primary goal of this work is to evaluate all positions on a notch in terms of crack initiation efficiently. Note, that the focus is on method development. Further improvements are possible by code optimizations, whereas the most time-consuming tasks are the FEM analyses. To investigate crack initiation on  $n_{pos}$  positions of a notch, Full FEA needs in total  $n_{pos} \cdot (n_{cracks} + 1)$  FEM analyses, whereas SLMM + AC only needs one FEM analysis for the voxel-model.



a) A strain of  $\varepsilon_{xx} = 0.1, \varepsilon_{yy} = 0.0, \gamma_{xy} = 0.0$  is applied to the model boundaries.



b) A strain of  $\varepsilon_{xx} = 0.0, \varepsilon_{yy} = 0.1, \gamma_{xy} = 0.0$  is applied to the model boundaries.



c) A strain of  $\varepsilon_{xx} = 0.0, \varepsilon_{yy} = 0.0, \gamma_{xy} = 0.1$  is applied to the model boundaries.

**Fig. 15.** Comparison of SLMM + AC results with Full FEA results. The left images depict the deformation of the notch and define and number critical positions. The failure indices are plotted in the middle images. The right images show the application of CC on the most critical positions.

The precomputation of the crack model needed by SLMM + AC does FEM analyses for each curvature  $c'$  of the crack model, for each deformation mode  $|\mathbb{D}|$  and for each introduced virtual crack length ( $n_{\text{cracks}} + 1$ ) including a zero-length crack. In total,  $n_{\text{curv}} \cdot |\mathbb{D}| \cdot (n_{\text{cracks}} + 1)$  FEM analyses are necessary. However, the precomputation is performed only once for one material and can be used again for other notch geometries.

The runtimes of SLMM + AC and Full FEA are further compared on a local PC with 16 GB RAM, an Intel® i5-6500 CPU with 4 cores, and 3.2

GHz clock rate. SLMM + AC and Full FEA is written in Anaconda Python 3.7 [30] and in Abaqus Python 2.7. Abaqus 2017 [29] is used for the Full FEA approach as well as the precomputation of the crack models. To avoid the dependency on the commercial Abaqus software during the application of SLMM + AC, the voxel model is implemented purely in Anaconda Python. SuperLU [31], provided by SciPy [32], solves the linear equation system, which arises from the voxel model.

The precomputation takes 1 h 46 min to compute the crack model in advance. SLMM + AC takes 2 min 48 s to simulate the voxel mesh with



**Table 3**

Validation of model parameters. In the model shown in Fig. 15a), analyses are performed at position (1) twice for each listed model parameter. Once with the default setting  $v_1$  and another time with the modified setting  $v_2$ . The maximum effect over the virtual path length for the max. principal stress and the incremental energy release rate are computed.

Parameter	$v_1$	$v_2$	$\Delta\sigma_1$	$\Delta G_{inc}$
$n_{fit,c}$	11	21	0.4%	0.7%
$n_{fit,u}$	28	58	0.4%	0.9%
$n_{cracks}$	10	5	–	14.2%
$D$	$D_{imp}$	$D_{all}$	2.5%	1.2%
$\varphi_{min}$	10	20	5.4%	17.21%
$l_{el,cm}/mm$	0.01	0.02	0.0%	0.6%
$n_{el,lm}$	$10^6$	$5 \cdot 10^5$	1.9%	3.9%
$l_{el,full}$	$\frac{v_1}{v_2} = 2$		0.1%	0.1%

$10^6$  voxel elements and 0.66 s to predict one position on the notch, whereas the Full FEA needs 2 min 50 s for each position.

### 3.3. Applicability of SLMM + AC

SLMM + AC should not only be efficient and accurate as discussed in the previous sections but should also be a versatile tool that can be used for different notches and different load cases. However, SLMM + AC computes results only up to a virtual crack length of  $a_n$  and therefore has some restrictions which we discuss in this section.

According to the CC and scaling laws the equation.

$$\underbrace{|c'| \frac{G'_{inc}(a'_c) \cdot E'}{(\sigma'_I(a'_c))^2}}_{simulation} = \underbrace{|c|}_{notch} \underbrace{\frac{G_c \cdot E}{\sigma_c^2}}_{l_{ch}} \quad (32)$$

is fulfilled for an initiating crack of length  $a_c = \lambda_1 \cdot a'_c$ . Both sides of equation (32) are dimensionless. The left side contains quantities corresponding to the precomputed crack model in the image space, whereas the right side contains the notch curvature  $c$  and Irwin's length  $l_{ch}$ , which is a material property composed of the fracture energy  $G_c$ , the Young's modulus  $E$ , and the strength  $\sigma_c$ .

SLMM + AC is only applicable if the length of the initiating crack  $a_c$  is not bigger than the maximum predictable virtual crack length  $a_n$ . Furthermore, the left side of equation (32) increases monotonically with

increasing  $a_c$ , if  $\sigma_1$  decreases and  $G_{inc}$  increases along the crack path. Consequently, evaluating the left side at the max. predictable virtual crack length  $a_c = a_n$  results in an upper bound which we call *applicability factor*.

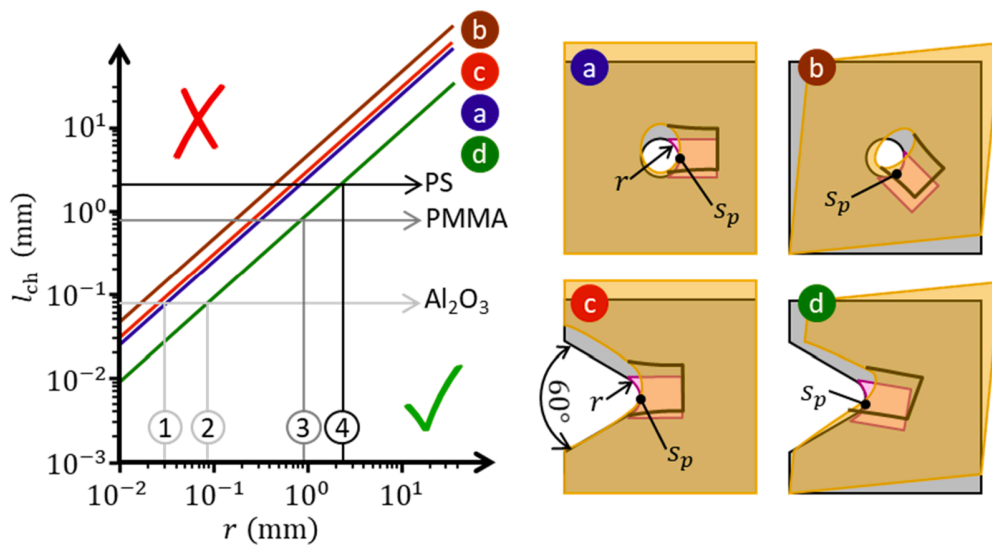
$$Apl_{max} = \frac{1}{r} \cdot \frac{G'_{inc}(a'_n) \cdot E'}{(\sigma'_I(a'_n))^2} \geq \frac{l_{ch}}{r} \quad (33)$$

We use the notch radius  $r \propto |c|^{-1}$  and Irwin's length  $l_{ch}$  instead of the notch curvature  $c$  and the material properties  $G_c$ ,  $E$ , and  $\sigma_c$ . Inequality (33) can be used to provide a lower limit for the notch radius  $r \geq l_{ch}/Apl_{max}$  and therefore, check if SLMM + AC is applicable to a problem.

We investigate the applicability for four problems: Two notch geometries each under two load cases. Fig. 16 depicts specimens with a circular hole in a) and b) as well as specimens with a blunted V-notch in c) and d). Uniaxial tension ( $\epsilon_{xx} = 0.0, \epsilon_{yy} = 0.1, \gamma_{xy} = 0.0$ ) is applied to the specimens a) and c), whereas b) and d) are loaded with pure shear ( $\epsilon_{xx} = 0.0, \epsilon_{yy} = 0.0, \gamma_{xy} = 0.1$ ). The radii  $r$  of the circular hole and the blunted V-notch are varied. The crack model is aligned to the position marked as  $s_p$  and has a length of  $l_{cm} = 1.95 \cdot r$  for the circular hole and  $l_{cm} = 2.15 \cdot r$  for the blunted V-notch. This corresponds to a curvature of  $c' = -0.6 \text{ mm}^{-1}$  in the virtual space. The crack model is placed horizontally on the right side of the notch for uniaxial tension a) and c), under  $45^\circ$  for specimen b) and  $10^\circ$  for specimen d), see Fig. 16. In the case of more complex notches, a smaller crack model in relation to the radius might be necessary to fit the notch surface. Then, the applicability factor  $Apl_{max}$  drops and must be recomputed.

The applicability factor is  $Apl_{max} = 2.53$  for a),  $Apl_{max} = 4.68$  for b),  $Apl_{max} = 3.06$  for c) and  $Apl_{max} = 0.964$  for d). The differences between the tension load for the circular hole a) and the blunted V-notch c) are rather small. The shear deformation for a circular hole corresponds to a rotated tension load, so the deviation between a) and b) is due to the rotation of the load. The shear load for the V-notch d) reduces  $Apl_{max}$  by a factor of about 3 compared to the tension load in c). For the computations, a Poisson's ratio of  $\nu = 0.35$  and plane stress conditions were used. However, we checked Poisson's ratios between  $\nu = 0.01$  and  $\nu = 0.49$  for specimen a) and observed only a small change of  $Apl_{max}$  between 2.29 and 2.96. A plane strain state leads for specimen a) to an applicability factor of  $Apl_{max} = 1.97$ .

The applicability factor can be used to check whether SLMM + AC is applicable to a problem. Therefore, Fig. 16 plots equation (33) in a



**Fig. 16.** The applicability of SLMM + AC is discussed with the help of four specimens a-d). SLMM + AC is applicable to a specimen with a certain radius  $r$  and a material with a certain Irwin's length  $l_{ch}$ , if the point  $(r, l_{ch})$  lies below the line corresponding to the right specimen and load case.

diagram where Irwin's length  $l_{ch}$  is plotted over the radius  $r$ . For the four cases a-d), separate lines are plotted. The region below a line is feasible for SLMM + AC. This allows finding the minimum notch radius  $r$  for a given specimen, material and load case. As depicted by line (1), a radius  $r$  of at least 0.03mm is required when a notch made of  $Al_2O_3$  under tension is investigated, whereas the specimen d) requires a radius  $r$  of at least 0.09mm (2). The notch radius must be greater than 0.9mm (3) for a PMMA specimen depicted in d) and greater than 2.3mm (4) if polystyrene is used instead.

#### 4. Conclusions

A fully automated and efficient method is proposed to predict crack initiation for arbitrary shaped notches and load cases using the Coupled Criterion (CC) in a 2D model. However, the notch must not contain a sharp edge or crack. The method splits a crack initiation analysis into two steps. In the first step, *crack models* are precomputed for one material and five unique deformation modes. In the second step, these pre-computed results are scaled to an actual notch using dimensional analysis, linear superposition, and meta-modeling. Furthermore, we propose a relaxation method for the boundary conditions to increase the accuracy of the incremental energy release rate prediction. The results are compared to those given by a *Full FEA* approach, which uses fully modeled finite element analyses.

Our main findings are:

1. The proposed method is tested with three load cases and can identify the most critical position on a notch with a complex shape. The prediction of the failure index at the most critical position is in all examples too conservative. The failure index is overestimated by up to 10.8% compared to predictions made by the Full FEA approach.
2. The proposed method is considerably more efficient than a Full FEA approach. Analyzing 50 positions on a notch takes 3 min 20 s with our novel method compared to 2 h 21 min with the Full FEA approach on a 4-core desktop computer.
3. Due to the multi-scale approach, only small initiating cracks can be predicted. Depending on the ratio of strength and fracture toughness, the notch geometry, and the load case, it is thus possible to investigate notches with a curvature radius above a limiting value. For a circular hole specimen under uniaxial tension, Irwin's length of the material must lie below 2.53 times the radius of the circular hole.

The proposed fully-automated and fast approach can be extended to 3d or can be used in models that feature crack growth as well. Furthermore, the method can help to model and thus optimize heterogeneous materials.

#### CRedit authorship contribution statement

**Matthias Retzl:** Conceptualization, Methodology, Software, Visualization, Validation, Data curation, Investigation, Writing – original draft, Writing – review & editing. **Martin Pletz:** Conceptualization, Methodology, Visualization, Writing – review & editing. **Clara Schuecker:** Conceptualization, Supervision, Writing – review & editing.

#### Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### References

- [1] R.M. Christensen, *The theory of materials failure*, 1st ed, Oxford University Press, Oxford, 2013.
- [2] T.L. Anderson, *Fracture mechanics: fundamentals and applications*, CRC Press, Boca Raton, 1991.
- [3] D. Taylor, Predicting the fracture strength of ceramic materials using the theory of critical distances, *Eng. Fract. Mech.* 71 (2004) 2407–2416, <https://doi.org/10.1016/j.engfracmech.2004.01.002>.
- [4] P. Cornetti, N. Pugno, A. Carpinteri, D. Taylor, Finite fracture mechanics: A coupled stress and energy failure criterion, *Eng. Fract. Mech.* 73 (2006) 2021–2033, <https://doi.org/10.1016/j.engfracmech.2006.03.010>.
- [5] D. Leguillon, Strength or toughness? A criterion for crack onset at a notch, *Eur. J. Mech. - ASolids*. 21 (2002) 61–72, [https://doi.org/10.1016/S0997-7538\(01\)01184-6](https://doi.org/10.1016/S0997-7538(01)01184-6).
- [6] D. Taylor, P. Cornetti, N. Pugno, The fracture mechanics of finite crack extension, *Eng. Fract. Mech.* 72 (2005) 1021–1038, <https://doi.org/10.1016/j.engfracmech.2004.07.001>.
- [7] D. Leguillon, Z. Yosibash, Crack onset at a v-notch. Influence of the notch tip radius, *Int. J. Fract.* 122 (2003) 1–21, <https://doi.org/10.1023/B:FRAC.000005372.68959.1d>.
- [8] Z. Yosibash, E. Priel, D. Leguillon, A failure criterion for brittle elastic materials under mixed-mode loading, *Int. J. Fract.* 141 (2006) 291–312, <https://doi.org/10.1007/s10704-006-0083-6>.
- [9] A. Carpinteri, P. Cornetti, N. Pugno, A. Sapora, D. Taylor, A finite fracture mechanics approach to structures with sharp V-notches, *Eng. Fract. Mech.* 75 (2008) 1736–1752, <https://doi.org/10.1016/j.engfracmech.2007.04.010>.
- [10] A. Sapora, P. Cornetti, A. Carpinteri, A Finite Fracture Mechanics approach to V-notched elements subjected to mixed-mode loading, *Eng. Fract. Mech.* 97 (2013) 216–226, <https://doi.org/10.1016/j.engfracmech.2012.11.006>.
- [11] J. Li, X.B. Zhang, A criterion study for non-singular stress concentrations in brittle or quasi-brittle materials, *Eng. Fract. Mech.* 73 (2006) 505–523, <https://doi.org/10.1016/j.engfracmech.2005.09.001>.
- [12] R. Romani, M. Bornert, D. Leguillon, R. Le Roy, K. Sab, Detection of crack onset in double cleavage drilled specimens of plaster under compression by digital image correlation – Theoretical predictions based on a coupled criterion, *Eur. J. Mech. - ASolids*. 51 (2015) 172–182, <https://doi.org/10.1016/j.euromechsol.2014.12.002>.
- [13] D. Leguillon, D. Quesada, C. Putot, E. Martin, Prediction of crack initiation at blunt notches and cavities – size effects, *Eng. Fract. Mech.* 74 (2007) 2420–2436, <https://doi.org/10.1016/j.engfracmech.2006.11.008>.
- [14] A. Doitrand, E. Martin, D. Leguillon, Numerical implementation of the coupled criterion: Matched asymptotic and full finite element approaches, *Finite Elem. Anal. Des.* 168 (2020), 103344, <https://doi.org/10.1016/j.finel.2019.103344>.
- [15] A. Sapora, A.R. Torabi, S. Etesam, P. Cornetti, Finite Fracture Mechanics crack initiation from a circular hole, *Fatigue Fract. Eng. Mater. Struct.* 41 (2018) 1627–1636, <https://doi.org/10.1111/ffe.12801>.
- [16] J. Li, D. Leguillon, Finite element implementation of the coupled criterion for numerical simulations of crack initiation and propagation in brittle materials, *Theor. Appl. Fract. Mech.* 93 (2018) 105–115, <https://doi.org/10.1016/j.tafmec.2017.07.010>.
- [17] P. Cornetti, M. Muñoz-Reja, A. Sapora, A. Carpinteri, Finite fracture mechanics and cohesive crack model: Weight functions vs. cohesive laws, *Int. J. Solids Struct.* 156–157 (2019) 126–136, <https://doi.org/10.1016/j.ijsolstr.2018.08.003>.
- [18] A. Doitrand, R. Estevez, D. Leguillon, Comparison between cohesive zone and coupled criterion modeling of crack initiation in rhombus hole specimens under quasi-static compression, *Theor. Appl. Fract. Mech.* 99 (2019) 51–59, <https://doi.org/10.1016/j.tafmec.2018.11.007>.
- [19] P. Cornetti, A. Sapora, Penny-shaped cracks by Finite Fracture Mechanics, *Int. J. Fract.* 219 (2019) 153–159, <https://doi.org/10.1007/s10704-019-00383-9>.
- [20] H. Böhm, *A short introduction to basic aspects of continuum micromechanics*, ILSB Rep. (2015).
- [21] M. Budinger, J.-C. Passieux, C. Gogu, A. Fraj, Scaling-law-based metamodelling for the sizing of mechatronic systems, *Mechatronics* 24 (2014) 775–787, <https://doi.org/10.1016/j.mechatronics.2013.11.012>.
- [22] L.L. Schumaker, P. Dierckx, Curve and Surface Fitting with Splines, *Math. Comput.* 63 (1994) 427, <https://doi.org/10.2307/2153590>.
- [23] A.G. de Miguel, I. Kaleel, M.H. Nagaraj, A. Pagani, M. Petrolo, E. Carrera, Accurate evaluation of failure indices of composite layered structures via various FE models, *Compos. Sci. Technol.* 167 (2018) 174–189, <https://doi.org/10.1016/j.compscitech.2018.07.031>.
- [24] E. Buckingham, On Physically Similar Systems; Illustrations of the Use of Dimensional Equations, *Phys. Rev.* 4 (1914) 345–376, <https://doi.org/10.1103/PhysRev.4.345>.
- [25] E. Martin, D. Leguillon, O. Sevecek, R. Bermejo, Understanding the tensile strength of ceramics in the presence of small critical flaws, *Eng. Fract. Mech.* 201 (2018) 167–175, <https://doi.org/10.1016/j.engfracmech.2018.06.021>.
- [26] D.C. Montgomery, G.C. Runger, *Applied statistics and probability for engineers*, 3rd ed, Wiley, New York, 2003.

- [27] R.F. Cook, B.R. Lawn, C.J. Fairbanks, Microstructure-Strength Properties in Ceramics: I, Effect of Crack Size on Toughness, *J. Am. Ceram. Soc.* 68 (1985) 604–615, <https://doi.org/10.1111/j.1151-2916.1985.tb16163.x>.
- [28] M.F. Ashby, H. Shercliff, D. Cebon, *Materials*, 4th edition, Elsevier, Cambridge, Oxford, 2019.
- [29] M. Smith, *ABAQUS/Standard User's Manual, Version 6.9*, Dassault Systèmes Simulia Corp, United States (2009).
- [30] *Anaconda Software Distribution*, n.d. <https://docs.anaconda.com/>.
- [31] X.S. Li, An Overview of SuperLU: Algorithms, Implementation, and User Interface, *Toms.* 31 (2005) 302–325.
- [32] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S.J. van der Walt, M. Brett, J. Wilson, K.J. Millman, N. Mayorov, A.R.J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, Í. Polat, Y.u. Feng, E.W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E.A. Quintero, C.R. Harris, A.M. Archibald, A.H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*, *Nat. Methods.* 17 (3) (2020) 261–272.



---

**Paper B**

# ConForce: Computation of Configurational Forces for FEM Results

Authors: Rettl, Matthias  
Frankl, Siegfried  
Pletz, Martin  
Tauscher, Markus  
Schuecker, Clara

SoftwareX

DOI: [10.1016/j.softx.2024.101718](https://doi.org/10.1016/j.softx.2024.101718)





Contents lists available at ScienceDirect

SoftwareX

journal homepage: [www.elsevier.com/locate/softx](http://www.elsevier.com/locate/softx)



# ConForce: Computation of configurational forces for FEM results

Matthias Retzl, Siegfried Martin Frankl, Martin Pletz\*, Markus Tauscher, Clara Schuecker

Chair of Designing Plastics and Composite Materials, Department of Polymer Engineering and Science, Montanuniversitaet Leoben, Austria

## ARTICLE INFO

**Keywords:**  
Finite element method  
Fracture mechanics  
Configurational forces  
Abaqus

## ABSTRACT

ConForce is a tool for evaluating nodal configurational forces (CF) from FEM results. ConForce supports 2D plane strain and 3D volume elements in a solver-independent manner. Furthermore, large displacements in static load cases are considered. ConForce accesses previously generated and compiled C code from Python via a C code binding. This compiled C code is provided for predefined element types, but can be regenerated for other element types. ConForce also includes an Abaqus plug-in with a graphical user interface. This Abaqus plug-in allows to easily add CF to the Abaqus output file.

## Metadata

Nr	Code metadata description	Please fill in this column
C1	Current code version	1.0.5
C2	Permanent link to code/repository used for this code version	<a href="https://github.com/mretzl/conforce">https://github.com/mretzl/conforce</a>
C3	Permanent link to reproducible capsule	
C4	Legal code license	MIT
C5	Code versioning system used	Git
C6	Software code languages, tools and services used	Python 3.7, Abaqus Python, C
C7	Compilation requirements, operating environments and dependencies	Operating systems: Windows 64-bit, Linux 64-bit For the plug-in: Abaqus 2017 to Abaqus 2023 For the Python package: NumPy, SymPy For the implementation of additional element types: GNU C-compiler
C8	If available, link to developer documentation/manual	<a href="https://conforce.readthedocs.io">https://conforce.readthedocs.io</a>
C9	Support email for questions	<a href="mailto:martin.pletz@unileoben.ac.at">martin.pletz@unileoben.ac.at</a>

computation method for the vectorial J-Integral of Budiansky and Rice [13]. CF relate the energy release with a change in the reference configuration [4] that may be a movement of an inhomogeneity or crack growth [3]. Furthermore, CF can be computed in the post-processing of a FE analysis without additional simulations.

Another benefit of CF as well as of the vectorial J-Integral is that in FM they predict the energy release rate and the direction of maximum energy release rate [14–16]. However, for curved cracks or mixed-mode loadings, the direction might be inaccurate and correction schemes are necessary [6,12].

Although, CF are popular, most commercial FEM codes do not provide an output of CF. An exception is Ansys [17] that provides a CF based evaluation of the crack tip J-Integral. We started developing ConForce for the paper of Frankl et al. [6] and make it publicly available in this work.

We consider two formulations of CF, which we call motion-based formulation (MBF) and displacement-base formulation (DBF). The MBF approach was presented by Mueller and Maugin [3] for elastic and non-linear elastic materials in the large strain framework. Based on the Helmholtz energy density  $\Psi$ , the identity matrix  $I$ , the deformation gradient  $F$ , and the first Piola-Kirchhoff stress tensor  $P$ , the Eshelby stress tensor or energy-momentum tensor

$$\Sigma^{(MBF)} = \Psi \cdot I - F^T P \quad (1)$$

is defined [18] in the reference configuration. According to Mueller and Maugin [3], the nodal CF for the  $i^{\text{th}}$  node and its corresponding shape function  $h_i$  are computed using two integrals:

## 1. Motivation and significance

Configurational forces (CF) are used in shape optimization [1], mesh optimization [2,3], and to study material inhomogeneities [4]. Furthermore, CF are a generalization of the scalar J-Integral of Rice [5] widely used in Fracture Mechanics (FM) [3,4,6–12] and an alternative

\* Corresponding author.

E-mail address: [martin.pletz@unileoben.ac.at](mailto:martin.pletz@unileoben.ac.at) (M. Pletz).

$$g_{\text{nodal},i}^{(\text{MBF})} = \underbrace{\int_{\partial\mathcal{B}} (\Sigma^{(\text{MBF})} N) \frac{\partial h_i}{\partial X} dS}_{g_{\text{nodal},i}^{(\text{MBF},S)}} - \underbrace{\int_{\mathcal{B}} \Sigma^{(\text{MBF})} : \frac{\partial h_i}{\partial X} dV}_{g_{\text{nodal},i}^{(\text{MBF},V)}} \quad (2)$$

The first integral yields  $g_{\text{nodal},i}^{(\text{MBF},S)}$  and is evaluated over the surface  $\partial\mathcal{B}$  of a body  $\mathcal{B}$ . It uses the Eshelby stress  $\Sigma^{(\text{MBF})}$ , the surface normal vector  $N$ , the  $i^{\text{th}}$  shape function  $h_i$ , and the reference position vector  $X$ . The second term  $g_{\text{nodal},i}^{(\text{MBF},V)}$  contains a volume integral over the body  $\mathcal{B}$ .

Following the work of Müller and Maugin [3], we compute only the  $g_{\text{nodal},i}^{(\text{MBF},V)}$  and neglect the surface integral  $g_{\text{nodal},i}^{(\text{MBF},S)} \approx 0$ , because the evaluation of the surface integral from FEM results introduces numerical errors. Neglecting  $g_{\text{nodal},i}^{(\text{MBF},S)}$  is valid for nodes not lying on a physical boundary. In FM,  $g_{\text{nodal},i}^{(\text{MBF},S)}$  is often neglected for nodes on crack faces, which might lead to inaccurate results [12].

Gurtin [19] describes the DBF approach for the evaluation of CF for infinitesimal deformations: First, a modified Eshelby relation

$$\Sigma^{(\text{DBF})} = \Psi \cdot J - \left( \frac{\partial U}{\partial X} \right)^{\top} P \quad (3)$$

is defined. Compared to Eq. (1), the deformation gradient is replaced by the gradient of the displacement vector  $U$ . Next, the nodal CF  $g_{\text{nodal},i}^{(\text{DBF},S)}$ ,  $g_{\text{nodal},i}^{(\text{DBF},V)}$ , and  $g_{\text{nodal},i}^{(\text{DBF})}$  are defined similar to Eq. (2). In absence of body forces, the CF of the two approaches are the same  $g_{\text{nodal},i}^{(\text{DBF})} = g_{\text{nodal},i}^{(\text{MBF})}$  [19]. Analogous to the MBF approach, the surface integral is neglected.

The MBF and DBF formulations of CF are valid for linear and non-linear elastic materials in the large strain framework. However, extensions for dynamics [4,11], plasticity [9,10], and the small strain framework [7] have been proposed.

## 2. Software description

ConForce can be used either as a Python package or as an Abaqus [20] plug-in. The Python package can be installed from PyPi [21]. The user can pass FEM results of any FEM code to a function that computes the nodal CF. The Abaqus plug-in reads the energy density, stresses, coordinates, and displacements from an Abaqus output database (ODB) file, computes the nodal CF, and writes them back to the ODB file.

### 2.1. Software architecture

As shown by Fig. 1, ConForce is split into three packages:

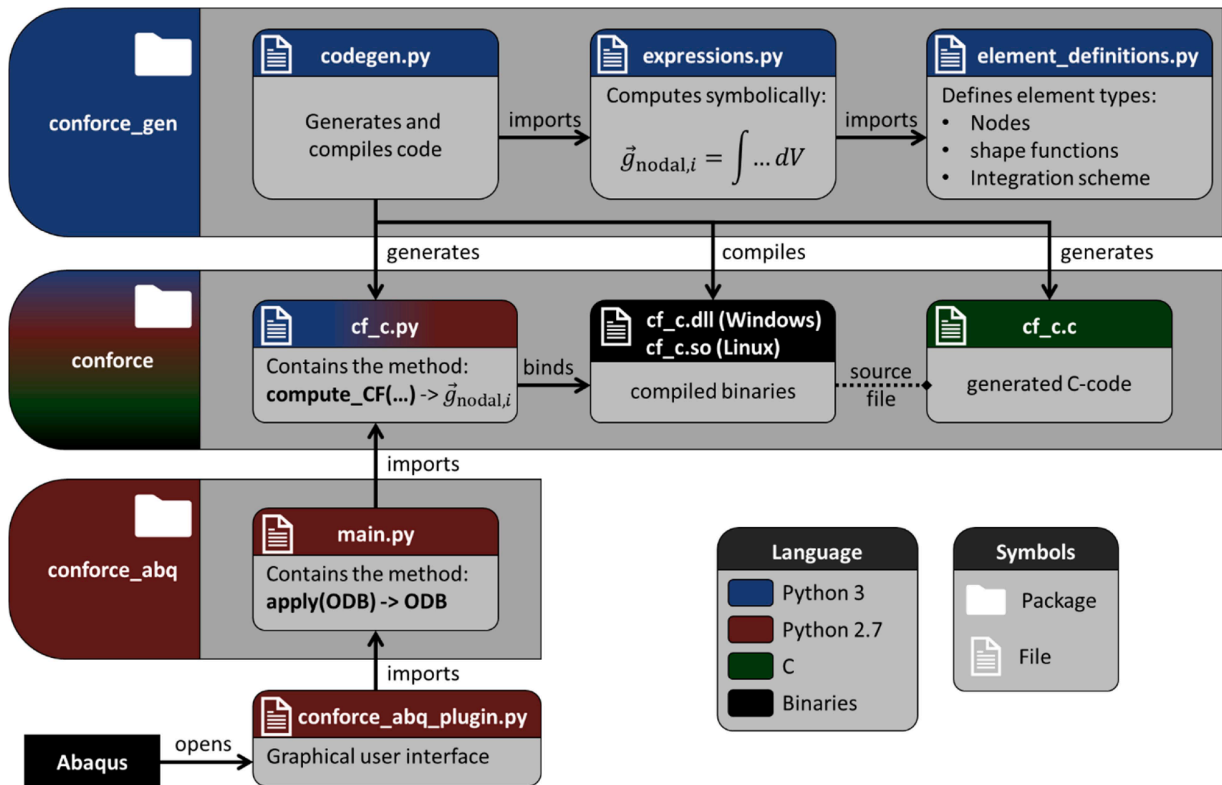
- `conforce_gen`: Symbolic computation, code generation and compilation of the package `conforce`.
- `conforce`: Methods for the computation of nodal CF.
- `conforce_abq`: Abaqus specific package for working with ODB files that is used by the Abaqus plug-in.

The packages run either in Python 2.7 (`conforce_abq/conforce`) or in Python 3 (`conforce_gen/conforce`). The package `conforce` is cross-compatible between Python 2.7 and Python 3. We use the deprecated Python 2.7 for the Abaqus-specific part, because Abaqus up to version 2023 uses Python 2.7. To develop and maintain ConForce as an up-to-date Python package, `conforce_gen` and `conforce` are executable in Python 3.

Additionally, a plug-in file `conforce_abq_plugin.py` is provided that contains code for a graphical user interface (GUI). This GUI can be opened in Abaqus by clicking on the toolbar entry Plug-ins -> Conforce.

#### Package ‘conforce\_gen’

ConForce already provides a number of element type



**Fig. 1.** Components of ConForce. The package `conforce_gen` generates the package `conforce`, which is cross-compatible between Python 2.7 and Python 3 and computes nodal CF. The Abaqus-specific package `conforce_abq` reads and writes nodal CF from and into the ODB file. The plug-in script `conforce_abq_plugin.py` contains code for a graphical user interface.

implementations for Windows 64-bit and Linux 64-bit. The `conforce_gen` package allows additional element types to be implemented or compiled for other operating systems. The package is used to generate the package `conforce`. The element types are defined in a reference space by the coordinates of their nodes and integration points, by their shape functions and their weights of the integration points. The module `element_definitions.py` contains these data. A new element type can be added by writing its element information into `element_definitions.py`. The module `expressions.py` uses this element information of `element_definitions.py` for the symbolic computation of the nodal CF, which is performed by the package `SymPy` [22]. From the symbolic computation, the module `codegen.py` generates the Python module `cf.c.py` and the C-code file `_cf.c.c` in the package `conforce`. Furthermore, `_cf.c.c` is compiled using the GNU `gcc` compiler into `_cf.c.dll` for Windows 64-bit or `_cf.c.so` for Linux 64-bit.

#### Package ‘conforce’

This package is cross-compatible between Python 2.7 and Python 3. The method `compute_CF` of the module `cf.c.py` provides an easy and fast computation of the nodal CF. The module `cf.c.py` binds to the binaries `_cf.c.dll` and `_cf.c.so` via the `ctypes` package. The computation itself is performed by the binaries efficiently.

#### Package ‘conforce\_abq’

This package contains Abaqus-specific Python 2.7 code. The module `main.py` provides a single function called `apply`, which reads displacements, stresses, and energy densities from an Abaqus ODB file and adds nodal CF to the same ODB file.

#### Plug-in script ‘conforce\_abq\_plugin.py’

The plug-in script `conforce_abq_plugin.py` defines a GUI, which can be opened in the Plug-in toolbar of Abaqus CAE. Once the user clicks the apply button in the GUI, the apply function of the package `conforce_abq` is called and nodal CF are computed for a selected Abaqus ODB file.

## 2.2. Software functionalities

ConForce can be used either as Abaqus plug-in or as a stand-alone Python package for Python environments with version 3.7 or higher.

### 2.2.1. Python package

The Python package for Python 3.7 and higher can be installed from PyPi with `pip`. `pip install conforce`

This installs the packages `conforce` and `conforce_gen` as described in Section 2.1 and already contains compiled binaries. ConForce is able to process the element types listed in Table 1. The element types are named according to the Abaqus convention [20]. Element names starting with “CPE” are continuum plane strain elements in  $d = 2$  dimensions. The prefix “C3D” stands for continuum  $d = 3$  dimensional elements.

The standalone version of ConForce requires the user to provide the node coordinates `X_at_nodes` in the reference configuration, the node displacements `U_at_nodes`, the energy densities `e_at_int_points`, and the

**Table 1**

Element types currently supported by ConForce. The brackets define the number of dimension  $d$ , the number of nodes  $n$ , and the number of integration points  $ips$  ordered as  $(d, n, ips)$ .

Integration order	Shape function	Linear		Quadratic	
		Full	Reduced	Full	Reduced
2D	Triangular	CPE3 (2, 3, 1)		CPE6 (2, 6, 3)	
	Quadrilateral	CPE4 (2, 4, 4)	CPE4R (2, 4, 1)	CPE8 (2, 8, 9)	CPE8R (2, 8, 4)
	Brick	C3D8 (3, 8, 4)	C3D8R (3, 8, 1)	C3D20 (3, 20, 27)	C3D20R (3, 20, 8)
3D	Tetrahedron	C3D4 (3, 4, 1)		C3D10 (3, 10, 4)	
	Triangular Prism	C3D6 (3, 6, 2)		C3D15 (3, 15, 9)	

Cauchy stress tensors at the integration points `S_at_int_points`. The detailed data structure is explained in the online documentation.

Code snippet 1 shows how to compute nodal CF with these data for one 2D element. First, the module `cf.c` from the package `conforce` is imported. Next, the method `compute_CF` is called. The nodal CF are computed using the MBF method for one element `el=1` of type “CPE4R” with  $d = 2$ ,  $n = 4$ , and  $ips=1$  as defined in Table 1. From the numbers `el`, `d`, `n`, and `ips`, the shapes of the arrays `X_at_nodes`, `U_at_nodes`, `e_at_int_points`, and `S_at_int_points` is derived as described in the code snippet. For example, `X_at_nodes` is an array of shape  $(el, n, d)$ . Note, that the order of the nodes and integration points inside the arrays must match the order defined in the Abaqus documentation for each element type. Unlike Abaqus, the Cauchy stress tensors in `S_at_int_points` are given as full (symmetric) stress tensors instead of vectorized tensors. This prevents confusion about the ordering of tensor components. Finally, the function returns the contribution of the element to the nodal CF. The user has to sum these contributions for each node. For cases with various element types, the method `compute_CF` must be called for each element type.

*Code snippet 1: Computation of nodal CF for an element with four nodes and one integration point in two dimensions.* from `conforce` import `cf.c` `cf.c`.  
`compute_CF`(

```

# supported methods are "mbf" or "dbf"
method="mbf",
# same element type names as Abaqus
element_type="CPE4R",
# coordinates of nodes
# in the reference configuration
# shape of array (el, n, d)
X_at_nodes=[[
    [0., 0.],
    [1., 0.],
    [1., 1.],
    [0., 1.],
]],
# displacement of nodes
# shape of array (el, n, d)
U_at_nodes=[[
    [0.0, 0.0],
    [0.1, 0.0],
    [0.1, 0.0],
    [0.0, 0.0],
]],
# energy densities at integration points
# in the reference configuration
# shape of array (el, ips)
e_at_int_points=[[10.]],
# Cauchy stress tensors at integration points
# shape of array (el, ips, d, d)
S_at_int_points=[[
    [100., 0.0],
    [0.0, 0.0]
]],
)
# Result array of shape (el, n, d):
# array([[50., -5.],
#        [-50., -5.],
#        [-50., 5.],
#        [50., 5.]])

```

### 2.2.2. Abaqus plug-in

The second way to use ConForce is the Abaqus plug-in provided with ConForce in the release folder. The downloaded folder needs to be placed into the plug-in directory of Abaqus CAE. A new entry named “Conf. Force” will appear in the “Plug-ins” toolbar the next time Abaqus CAE is started. Clicking on this “Conf. Force” entry will open the GUI



shown in Fig. 2.

The GUI allows to select an ODB file opened in the current Abaqus window. The simulation should use the large strain framework by setting the NLGEOM flag to ON. The ODB file must contain at least the following field outputs:

- displacements (U)
- stresses (S)
- energy densities (such as SENER)

In addition, the ODB must have been opened with the read-only check box unchecked, such that the new output can be written to the ODB file by the plug-in.

The computation method defines if the MBF or DBF formulation is used. The next text field considers the field output name of the energy density. ConForce supports elasticity and should consider the elastic strain energy density “SENER”. However, Kolednik et al. [9] suggest adding the plastic energy density “SENER+PENER” to account for small strain plasticity. The next section in the GUI is called “field output” and states which output should be written to the ODB file. In the case shown in Fig. 2, only the nodal CF are written to the ODB file. After the computation, they are available for visualization in Abaqus by changing the field output to “CONF\_FORCE”. Fig. 3 depicts an example of nodal CF as a vector plot in Abaqus. Note, that biquadratic elements are used and hence the mid-side nodes have higher CF than the corner nodes of an element.

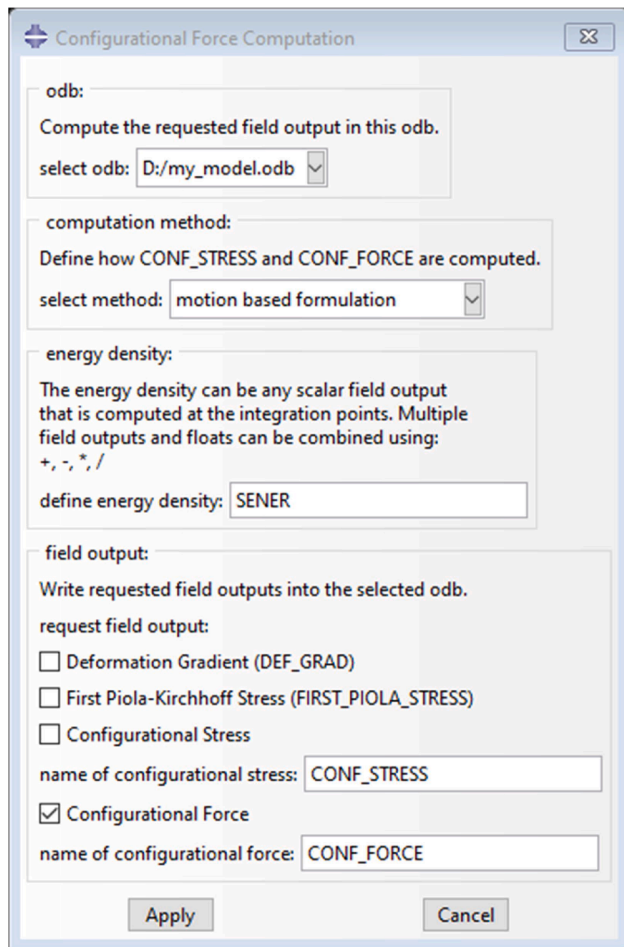


Fig. 2. Graphical user interface of the Abaqus plug-in.

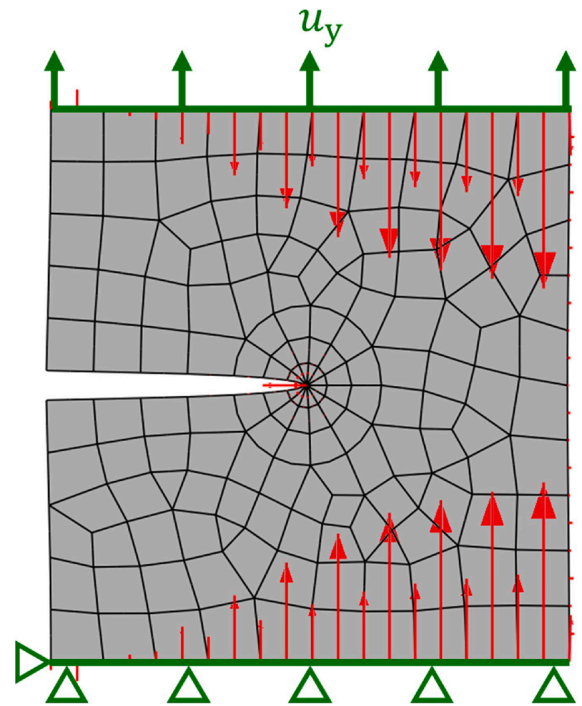


Fig. 3. Schematic vector plot of computed nodal  $g_{nodal}^{(MBF, V)}$  (red) on a model of a crack in Abaqus under a tensile load  $u_y$ .

### 3. Illustrative examples

#### 3.1. Two-phase bar

The two-phase bar shown in Fig. 4 is used for the validation of evaluated interface CF with literature [4]. The bar has a height  $h = 10$  mm, a length  $l = 20$  mm, and a thickness  $t = 1$  mm. An interface is located at  $l_1 = 10$  mm. The left side is fully constrained and along the right edge a displacement  $u = 0.1$  mm is applied. The bar consists of two bilinear plane strain elements. The left element's Young's modulus of  $E_1 = 210$  GPa is twice the Young's modulus of the right element ( $E_2 = 105$  GPa). Hence, the strain energy increases when the interface is shifted to the right. Transversal strain is neglected by setting the Poisson's ratio to zero ( $\nu = 0$ ).

Kolling and Mueller [4] provide an analytical solution of the energy release rate for a horizontal shift of this interface:

$$G = \frac{2 u^2 E_1 E_2 h t (E_1 - E_2)}{l^2 (E_1 + E_2)^2} \approx 11.67 \frac{\text{J}}{\text{m}} \quad (4)$$

The energy release rate  $G$  states, that if  $l_1$  increases by a small  $\Delta x$ , the

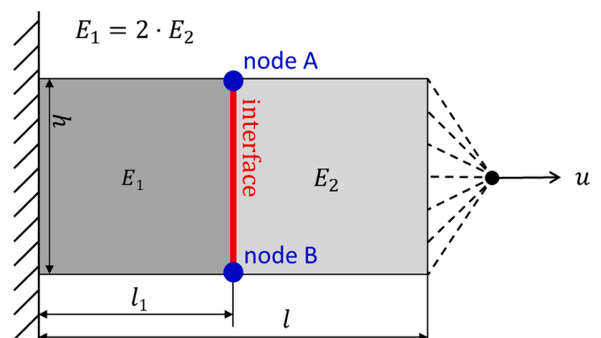


Fig. 4. Two-phase bar with a stiffer material  $E_1$  and a less stiff material  $E_2$ .

strain energy increases by  $\Delta x \cdot 11.67 \text{ J/m}$ . For the validation, the two-phase bar is simulated in Abaqus. ConForce computes horizontal nodal  $g_{x, \text{nodal}}^{(\text{MBF}, \text{V})}$  of 5.86 N in nodes A and B. The sum of those forces is

$$G \approx 5.86 \text{ N} + 5.86 \text{ N} = 11.72 \frac{\text{J}}{\text{m}}, \quad (5)$$

which corresponds well to the analytical result.

### 3.2. Mode I and mixed-mode loaded crack

This example considers a circular model that is depicted in Fig. 5. The model contains a crack with a crack tip that is located at the origin of the model in the center of the disk. The model is loaded by a displacement field associated with stress intensity factors  $K_I$  and  $K_{II}$ . Anderson [23] provides the corresponding displacement field:

$$\mathbf{u}(\varphi, K_I, K_{II}) = \mathbf{u}_{K_I}(\varphi, K_I) + \mathbf{u}_{K_{II}}(\varphi, K_{II})$$

$$\text{with } \begin{cases} \mathbf{u}_{K_I}(\varphi, K_I) = \frac{K_I}{2G} \sqrt{\frac{R}{2\pi}} \begin{pmatrix} \cos\left(\frac{\varphi}{2}\right) \left(\kappa - 1 + 2\sin^2\left(\frac{\varphi}{2}\right)\right) \\ \sin\left(\frac{\varphi}{2}\right) \left(\kappa + 1 - 2\cos^2\left(\frac{\varphi}{2}\right)\right) \end{pmatrix} \\ \mathbf{u}_{K_{II}}(\varphi, K_{II}) = \frac{K_{II}}{2G} \sqrt{\frac{R}{2\pi}} \begin{pmatrix} \sin\left(\frac{\varphi}{2}\right) \left(\kappa + 1 + 2\cos^2\left(\frac{\varphi}{2}\right)\right) \\ -\cos\left(\frac{\varphi}{2}\right) \left(\kappa - 1 - 2\sin^2\left(\frac{\varphi}{2}\right)\right) \end{pmatrix} \end{cases} \quad (6)$$

$$\kappa = 3 - 4\nu$$

This  $\mathbf{u}(\varphi, K_I, K_{II})$  is applied at the outer ring with radius  $R = 50 \text{ mm}$  in the circular model. The shear modulus is defined by  $G = E / (2 + 2\nu)$  with the Young's modulus  $E = 210 \text{ GPa}$  and the Poisson's ratio  $\nu = 0.3$  for a linear elastic material. The model has a thickness of  $t = 1 \text{ mm}$ .

The model is meshed using fully-integrated bilinear plane strain elements and uses nonlinear geometry. The mesh size at the outer edge is 8 mm and is refined towards the region  $\mathcal{A}$ . This region is a square with an edge length of 2.8 mm and a mesh size of 0.05 mm.

#### 3.2.1. Pure mode I loading

For this load case, we choose  $K_I = 20 \text{ MPa}\sqrt{\text{m}}$  and  $K_{II} = 0 \text{ MPa}\sqrt{\text{m}}$ . For a plane strain state, the applied energy release rate  $G_{\text{appl}}$  calculates as  $K_I^2 \cdot (1 - \nu^2) / E = 1733.33 \text{ J/m}^2$  [23]. Two simulations are performed. In the first simulation,  $\mathbf{CF}_{\mathcal{A}} = \sum_{\mathcal{A}} \mathbf{g}_{\text{nodal}}^{(\text{MBF}, \text{V})} = [-1738.31, 0.00] \text{ J/m}^2$  and the J-Integral  $G_I = 1738.66 \text{ J/m}^2$  are computed inside  $\mathcal{A}$ . The energy release rate according to ConForce is  $G_{\text{CF}} = |\mathbf{CF}_{\mathcal{A}}|_2 = 1738.31 \text{ J/m}^2$ . We chose a generously-sized region  $\mathcal{A}$  to account not only for CF at the crack tip, but also for so-called spurious configurational forces in the vicinity of the crack tip [24]. In the second simulation, one node is closed and hence

the crack tip moves leftwards by the distance of one element edge length  $\Delta a = 0.05 \text{ mm}$ . From the associated change in the strain energy  $\Delta \Pi$ , the energy release rate can be written as  $G_{\Delta \Pi} = \Delta \Pi / (t \cdot \Delta a) = 1735.24 \text{ J/m}^2$ . The energy release rate  $G_{\text{CF}}$  deviates less than 0.3% from  $G_{\text{appl}}$  and also agrees well with the  $G_{\Delta \Pi}$  and  $G_I$ .

#### 3.2.2. Mixed-mode loading

For this load case, we choose  $K_I = 20 \text{ MPa}\sqrt{\text{m}}$  and  $K_{II} = 10 \text{ MPa}\sqrt{\text{m}}$ . The nodal  $\mathbf{g}_{\text{nodal}}^{(\text{MBF}, \text{V})}$  are summarized to the resulting configurational forces  $\mathbf{CF}_3 = [-2149, 1686] \text{ J/m}^2$  and  $\mathbf{CF}_{15} = [-2155, 1697] \text{ J/m}^2$  for the regions  $\mathcal{E}_3$  and  $\mathcal{E}_{15}$ . Additionally, Abaqus is used to compute the vectorial J-Integral for both regions as  $\mathbf{J}_3 = [2147, -1689] \text{ J/m}^2$  and  $\mathbf{J}_{15} = [2154, -1699] \text{ J/m}^2$ . The energy release rates are estimated as the magnitude of the resulting configurational forces and vectorial J-Integrals. This results in  $G_{3, \text{CF}} = |\mathbf{CF}_3|_2 = 2731 \text{ J/m}^2$ ,  $G_{15, \text{CF}} = |\mathbf{CF}_{15}|_2 = 2743 \text{ J/m}^2$ ,  $G_{3, \text{J}} = |\mathbf{J}_3|_2 = 2732 \text{ J/m}^2$ , and  $G_{15, \text{J}} = |\mathbf{J}_{15}|_2 = 2743 \text{ J/m}^2$ .

The results show, that the CF do not vary when the size of the evaluation region is increased by a factor of five. However, this is not always the case. Schmitz and Ricoeur [12] provide a correction method when CF vary with the size of the evaluation region. Furthermore, the results computed by the Abaqus J-Integral and ConForce are almost identical up to the fourth digit. Except for the flipped signs, which is intended.

## 4. Impact

Though configurational forces can contribute in many fields such as inhomogeneous fracture mechanics or topology optimization, their use is limited today. The results of FEM models provide all the information, but implementing the evaluating of CF is an obstacle to research groups that would need them. ConForce provides an efficient CF implementation that has been tested and documented to such groups. Therefore, ConForce facilitates the use of configurational forces.

## 5. Conclusions

A Python package and Abaqus plug-in for the computation of nodal configurational forces is presented. This allows an efficient and accurate evaluation of configurational forces from FEM results. ConForce supports common 3D and plane-strain 2D elements. For each element type the equations are solved symbolically and then written in C-code, which is compiled and is accessible in Python through a C-code binding. This makes ConForce fast enough to be applied to FEM models with many thousand nodes.

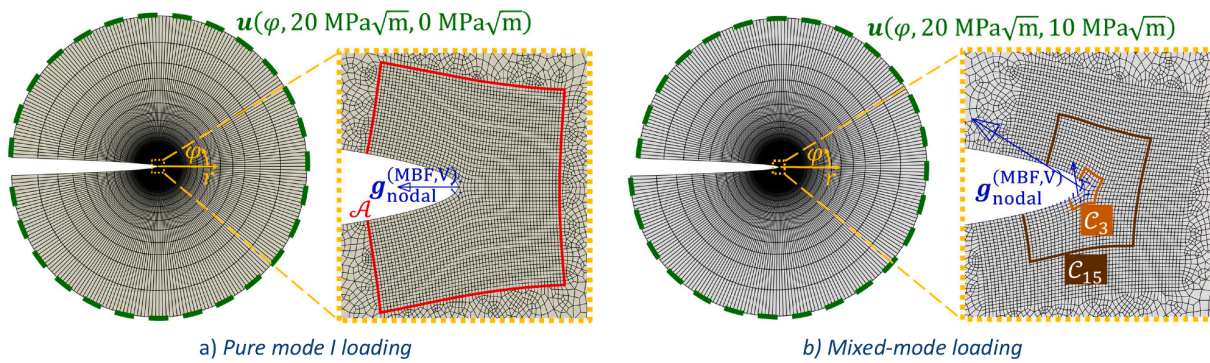


Fig. 5. Two load cases of a crack model. At the dashed green boundary, a displacement field  $u(\varphi, K_I, K_{II})$  is applied. The nodal  $g_{\text{nodal}}^{(\text{MBF}, \text{V})}$  are summed a) inside the region  $\mathcal{A}$  or b) inside the regions  $\mathcal{E}_3$  and  $\mathcal{E}_{15}$ . The model's deformations are magnified by a factor of 100 for visualization purposes.

### CRedit authorship contribution statement

**Matthias Rettl:** Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Siegfried Martin Frankl:** Formal analysis, Methodology, Writing – review & editing. **Martin Pletz:** Methodology, Visualization, Writing – review & editing. **Markus Tauscher:** Methodology, Software, Writing – review & editing. **Clara Schuecker:** Methodology, Supervision, Writing – review & editing.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Data availability

Data will be made available on request.

### References

- [1] Materna D, Barthold FJ. On variational sensitivity analysis and configurational mechanics. *Comput. Mech.* 2008;41:661–81. <https://doi.org/10.1007/s00466-007-0223-5>.
- [2] Mueller R, Gross D, Maugin GA. Use of material forces in adaptive finite element methods. *Comput. Mech.* 2004;33:421–34. <https://doi.org/10.1007/s00466-003-0543-z>.
- [3] Mueller R, Maugin GA. On material forces and finite element discretizations. *Comput. Mech.* 2002;29:52–60. <https://doi.org/10.1007/s00466-002-0322-2>.
- [4] Kolling S, Mueller R. On configurational forces in short-time dynamics and their computation with an explicit solver. *Comput. Mech.* 2005;35:392–9. <https://doi.org/10.1007/s00466-004-0627-4>.
- [5] Rice JR. A path independent integral and the approximate analysis of strain concentration by notches and cracks. *J. Appl. Mech.* 1968;35:379–86. <https://doi.org/10.1115/1.3601206>.
- [6] Frankl SM, Pletz M, Schuecker C. Improved concept for iterative crack propagation using configurational forces for targeted angle correction. *Eng. Fract. Mech.* 2022; 266:108403. <https://doi.org/10.1016/j.engfracmech.2022.108403>.
- [7] Miehe C, Gürses E, Birkle M. A computational framework of configurational-force-driven brittle fracture based on incremental energy minimization. *Int. J. Fract.* 2007;145:245–59. <https://doi.org/10.1007/s10704-007-9078-1>.
- [8] Ochensberger W, Kolednik O. Physically appropriate characterization of fatigue crack propagation rate in elastic–plastic materials using the J -integral concept. *Int. J. Fract.* 2015;192:25–45. <https://doi.org/10.1007/s10704-014-9983-z>.
- [9] Kolednik O, Schöngrundner R, Fischer FD. A new view on J-integrals in elastic–plastic materials. *Int. J. Fract.* 2014;187:77–107. <https://doi.org/10.1007/s10704-013-9920-6>.
- [10] Simha N, Fischer F, Shan G, Chen C, Kolednik O. J-integral and crack driving force in elastic–plastic materials. *J. Mech. Phys. Solids* 2008;56:2876–95. <https://doi.org/10.1016/j.jmps.2008.04.003>.
- [11] Özenç K, Chinarian G, Kaliske M. A configurational force approach to model the branching phenomenon in dynamic brittle fracture. *Eng. Fract. Mech.* 2016;157: 26–42. <https://doi.org/10.1016/j.engfracmech.2016.02.017>.
- [12] Schmitz K, Ricoeur A. Theoretical and computational aspects of configurational forces in three-dimensional crack problems. *Int. J. Solids Struct.* 2023;282:112456. <https://doi.org/10.1016/j.ijsolstr.2023.112456>.
- [13] Budiansky B, Rice JR. Conservation laws and energy-release rates. *J. Appl. Mech.* 1973;40:201–3. <https://doi.org/10.1115/1.3422926>.
- [14] Judt PO, Ricoeur A. Crack growth simulation of multiple cracks systems applying remote contour interaction integrals. *Theor. Appl. Fract. Mech.* 2015;75:78–88. <https://doi.org/10.1016/j.tafmec.2014.11.001>.
- [15] Guo Y, Li Q. Material configurational forces applied to mixed mode crack propagation. *Theor. Appl. Fract. Mech.* 2017;89:147–57. <https://doi.org/10.1016/j.tafmec.2017.02.006>.
- [16] Ma L, Korsunsky AM. On the use of vector j-integral in crack growth criteria for brittle solids. *Int. J. Fract.* 2005;133:L39–46. <https://doi.org/10.1007/s10704-005-0631-5>.
- [17] DeSalvo Gabriel J, Swanson John A. ANSYS engineering analysis system user's manual. Houston, Pa: Swanson Analysis Systems; 1985. <https://search.library.wisc.edu/catalog/999581007202121>.
- [18] Eshelby JD. The elastic energy-momentum tensor. *J. Elast.* 1975;5:321–35. <https://doi.org/10.1007/BF00126994>.
- [19] Gurtin ME. *Configurational forces as basic concepts of continuum physics*. New York, NY: Springer New York; 2000. <https://doi.org/10.1007/b97847>.
- [20] Abaqus, (2017). <https://www.3ds.com/simulia>.
- [21] Python Package Index - PyPI, (2024). <https://pypi.org/> (accessed January 2, 2024).
- [22] Meurer A, Smith CP, Paprocki M, Čertík O, Kirpichev SB, Rocklin M, Kumar A, Ivanov S, Moore JK, Singh S, Rathnayake T, Vig S, Granger BE, Muller RP, Bonazzi F, Gupta H, Vats S, Johansson F, Pedregosa F, Curry MJ, Terrel AR, Roučka Š, Saboo A, Fernando I, Kulal S, Cimrman R, Scopatz A. SymPy: symbolic computing in Python. *PeerJ Comput. Sci.* 2017;3. <https://doi.org/10.7717/peerj-cs.103>.
- [23] Anderson TL. *Fracture mechanics: fundamentals and applications*. CRC Press; 1991.
- [24] Denzer R, Barth FJ, Steinmann P. Studies in elastic fracture mechanics based on the material force method. *Int. J. Numer. Methods Eng.* 2003;58:1817–35. <https://doi.org/10.1002/nme.834>.



---

**Paper C**

# Evaluation of Combinatorial Algorithms for Optimizing Highly Nonlinear Structural Problems

Authors: Rettl, Matthias  
Pletz, Martin  
Schuecker, Clara

Materials & Design

DOI: [10.1016/j.matdes.2023.111958](https://doi.org/10.1016/j.matdes.2023.111958)

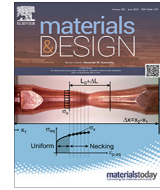




Contents lists available at ScienceDirect

# Materials & Design

journal homepage: [www.elsevier.com/locate/matdes](http://www.elsevier.com/locate/matdes)



## Evaluation of combinatorial algorithms for optimizing highly nonlinear structural problems

Matthias Rettl, Martin Pletz\*, Clara Schuecker

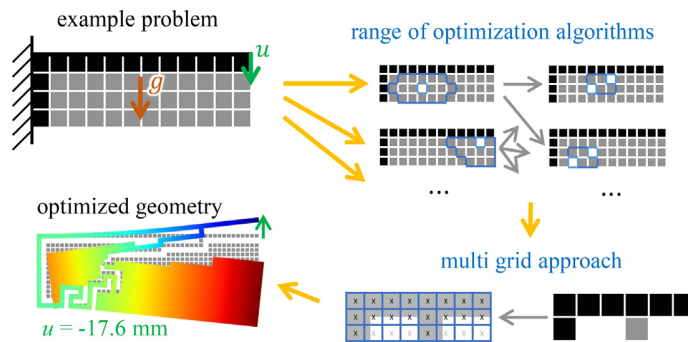
Chair of Designing Plastics and Composite Materials, Department of Polymer Engineering and Science, Montanuniversitaet Leoben, Austria



### HIGHLIGHTS

- Existing and newly developed optimization algorithms for a highly nonlinear problem are evaluated.
- The algorithms are evaluated in terms of efficiency and optimization results.
- The optimization results are compared to a global optimum.
- A new optimization approach that includes a multi grid approach yields promising results.

### GRAPHICAL ABSTRACT



### ARTICLE INFO

**Article history:**  
 Received 16 December 2022  
 Revised 14 April 2023  
 Accepted 22 April 2023  
 Available online 28 April 2023

**Keywords:**  
 Topology optimization  
 Global optimization  
 FEM  
 Computer-aided design

### ABSTRACT

Optimizing highly nonlinear structural problems can be very challenging due to the large number of parameters. Classical compliance minimization does not work for such problems. Common optimization algorithms also do not find good solutions. This work evaluates both commonly used optimization algorithms and algorithms not yet used in topology optimization. The algorithms are evaluated using a simple nonlinear problem: minimizing the end displacement of a cantilever beam fixed on one side and loaded by gravity. The global optimum for a coarse mesh grid is computed by simulating nearly 60 million possible topology designs using a Brute-Force search. We use this benchmark to evaluate the computational cost and objective values of known and newly developed optimization methods. The known methods are binary-coded Genetic Algorithm, Simulated Annealing, and Free Shape Optimization. The Reduced Variable Neighborhood Search (RVNS) has not yet been applied to topology optimization. We provide two implementations of RVNS: Breadth-First Search with a limited search depth (BFSL) and with an optional restriction for the size of the simultaneously modified area (TBFSL). According to the benchmark, TBFSL is the most efficient approach. For the optimization on a finer mesh grid, TBFSL is combined with a multi-grid approach to further increase efficiency.

© 2023 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

### 1. Introduction

For many engineering problems it is important to optimize the geometry of a component to achieve the desired behavior. In topol-

ogy optimization, the geometry in a given design space is optimized in terms of an objective function [1,2,3,4]. In many cases, the objective function is compliance, which should be minimized for a fixed volume. Virtual densities for each element are used as variables. If the variables are continuous, it is possible to compute the gradient, which is required for many commonly used optimization methods such as sequential quadratic programming [5]. How-

\* Corresponding author.

E-mail address: [martin.pletz@unileoben.ac.at](mailto:martin.pletz@unileoben.ac.at) (M. Pletz).

ever, the final design requires a clear distinction between material and voids. This can be done by penalizing intermediate densities and introducing a threshold. Another approach uses discrete variables that can only be set to either zero or one. A disadvantage of using discrete variables is that a gradient cannot be computed by deriving the objective function. Therefore, non-gradient-based methods such as Genetic Algorithm (GA) [6,7,8,9], Simulated Annealing (SA) [10,11,12], or Particle Swarm Optimization (PSO) [13] need to be used. Recently, neural networks have gained attention as surrogate models to further accelerate the optimization process [14].

Shape optimization [15,16] is an alternative approach that selects either a linear combination of shape functions (such as splines, NURBS, etc.) that represent the geometry, or nodes lying on the surface of a geometry that are moved directly [17,18]. Note that shape optimization cannot change the topology but only the outer shape of the structure.

Existing topology and shape optimization approaches work well for objective functions such as the compliance, but they do not perform well for highly nonlinear objective functions containing many local optima. The goal of this work is to develop optimization algorithms for such complex problems and to compare them with existing algorithms.

As an example for structural optimization, a cantilever beam loaded by its own weight is used. The objective is to minimize the vertical displacement of the topmost right point of the beam, while the left side is fixed. Changing the geometry strongly affects the load on the structure, making the problem highly nonlinear with many local optima.

The example of a dead weight-loaded beam is studied using existing optimization methods such as GA, SA, and Free Shape (FS). GA and SA are common methods for combinatorial optimization and FS is a shape optimization that describes each point of the geometry boundary independently. Furthermore, combinatorial optimization methods are proposed that have not yet been used in topology optimization. They are based on Breadth-First Search and Reduced Variable Neighborhood Search [19]. The Breadth-First Search first varies many elements individually before varying combinations of a few elements. However, since the Breadth-First Search would generate too many design proposals, restrictions on the search are required. Therefore, the newly developed optimization methods Breadth-First Search with Limited depth (BFSL) and Truncated Breadth-First Search with Limited depth (TBFSL) limit the number of design proposals. For a rather coarse mesh, these methods are evaluated against a Brute-Force (BF) computation of all existing beam designs.

To optimize a finer mesh, the TBFSL method is embedded in a Multi-Grid (MG) approach, which first optimizes a coarse mesh, and then transfers the result to a finer mesh, which is optimized again, and so on. For the best geometry found on a fine mesh, the vertical displacement of the topmost right point of the beam is negative, so this point moves upward.

## 2. Methods

### 2.1. Example problem definition

In this work, the geometry of a cantilever beam under dead load is optimized. Fig. 1 shows the design space of the beam, which has a length of 120 mm and a height of 40 mm. The objective is to minimize the endpoint displacement  $u$  at the top right point. The grid elements in Fig. 1 contain either material or no material. All nodes at the left edge are fixed. A gravity  $g$  acts as a body force. The material and load parameters are listed in Table 1. The cantilever beam is simulated using the Finite Element method. Linear geometry and

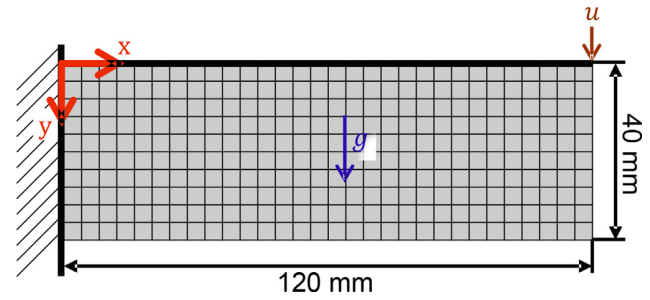


Fig. 1. Example problem definition for the optimization: The endpoint displacement  $u$  of the cantilever beam loaded by gravity  $g$  to be minimized in the following.

Table 1  
Material and load parameters used in the example problem.

Young's modulus	$E$	200MPa
Poisson's ratio	$\nu$	0.45
Density	$\rho$	1000kg/m <sup>3</sup>
Gravity constant	$g$	9.81m/s <sup>2</sup>

a plane stress state with a linear elastic material are assumed. Self-contact is not considered. Isoparametric rectangular eight-node elements with reduced integration are used. The linear solver uses a preconditioned conjugate gradient method [20] with a symmetric successive over-relaxation [21] preconditioner and an over-relaxation factor of 1.6. The model setup, the FEM code, the linear solver, and the optimizers were written in-house and implemented in JAVA.

Removing material results in less bending stiffness, but also less gravitational force on the beam. This nonlinear load makes it difficult to find an optimal design. In addition, the design must satisfy some constraints because not all combinations of filled and empty grid elements are valid. Designs that contain at least one freely movable element result in a singular stiffness matrix that cannot be solved and are therefore invalid. The black and gray elements in Fig. 2 represent a valid design in the design space. The red elements are not valid, because they can move or rotate freely. The design space

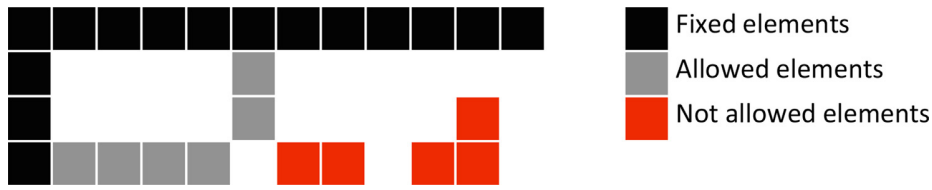
$$\mathbb{D} \left( N_{-el} \right) = \left\{ \mathbf{d} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & d_{22} & \dots & d_{2N_{el,x}} \\ \dots & \dots & \dots & \dots \\ 1 & d_{N_{el,y}2} & \dots & d_{N_{el,y}N_{el,x}} \end{bmatrix} \mid d_{ij} \in \{0, 1\} \wedge \text{valid}(\mathbf{d}) \right\} \quad (1)$$

is the set of all valid designs  $\mathbf{d}$  with a certain number of cells per row and per column  $N_{-el} = (N_{el,x}, N_{el,y})$ . A design  $\mathbf{d} \in \mathbb{D} \left( N_{-el} \right)$  directly assigns the value  $d_{ij}$  to the element in the  $i$ -th row and  $j$ -th column. A value of  $d_{ij} = 1$  turns on the  $i, j$ -th element, while a value of  $d_{ij} = 0$  turns off the  $i, j$ -th element. The left column and the top row of elements always have a value of 1 and are thus turned on. For finer meshes, the size of the fixed left and top elements decreases, and the variable design space grows. Since turned-off elements do not contribute to the gravity load, the load vector depends on the design.

The best design

$$\mathbf{d}_{\min} \in \arg \min_{\mathbf{d} \in \mathbb{D} \left( N_{-el} \right)} (\text{objective}(\mathbf{d})) \quad (2)$$

minimizes the  $y$ -displacement  $u$  at the upper right endpoint. An objective function  $u = \text{objective}(\mathbf{d})$  simulates the design  $\mathbf{d}$  and computes the endpoint displacement  $u$ .



**Fig. 2.** Example design in the design space  $\mathbb{D}((3, 11))$ . The topmost elements and leftmost elements are always turned on. Other elements are either turned on ( $d_{ij} = 1$ ) or off ( $d_{ij} = 0$ ). Elements that would result in a free body motion and rotation are not allowed.

The exact solution  $\mathbf{d}_{\min}$  of Eq. (2) is called the global optimum. No other design in the design space can have a smaller endpoint displacement than this exact solution. However, the global optimum is often extremely difficult to find. Many optimization algorithms use information gained from previous computations to find a possible good design. These algorithms tend to find only a locally optimal design that is surrounded by designs with a higher endpoint displacement. To increase the probability of finding the global optimum, the entire design space should be explored uniformly. A number of optimization methods are defined below, which are then used to optimize the beam geometry. Appendix A contains the detailed algorithms of the optimization methods.

## 2.2. Common methods in topology optimization

### 2.2.1. Brute-Force search (BF)

The Brute-Force BF  $\binom{N}{-el}$  search, see Appendix A.1, exhaustively explores the entire design space  $\mathbb{D}\left(\binom{N}{-el}\right)$  exhaustively by evaluating every possible design in it, and then selects the best design  $\mathbf{d}_{\min}$ . Thus, BF guarantees to find a global optimum. However, the number of simulations  $N_{\text{sim}} = \mathcal{O}(2^{N_{\text{var}}})$  grows exponentially with the number of variables  $N_{\text{var}} = (N_{\text{el},x} - 1) \cdot N_{\text{el},y} - 1$ . For this reason, global optima are rarely reported in the literature for structural optimization problems. For small problems, sophisticated branch-and-cut approaches can find a global optimum [5]. However, branch-and-cut approaches require a problem-specific formulation and they cannot be easily applied to other problems. In contrast, BF is a general-purpose method that can be applied to any problem whose input arguments are enumerable and finite. BF is a common method in cryptography for breaking weak ciphers [22], but it is rarely used in topology optimization.

### 2.2.2. Random Sampling (RS)

Random Sampling RS  $\left(\binom{N}{-el}, u_{\text{lim}}\right)$ , see Appendix A.2, randomly and uniquely samples designs  $\mathbf{d} \in \mathbb{D}\left(\binom{N}{-el}\right)$  until it finds a design with an endpoint displacement  $u$  less than or equal to an endpoint displacement limit  $u_{\text{lim}}$ . Like BF, RS does not use any information gained from the computation history, such as good design patterns or gradients. Unlike BF, RS guarantees a globally optimal design only if the limit  $u_{\text{lim}}$  is equal to the globally optimal endpoint displacement.

### 2.2.3. Genetic algorithm (GA)

Since BF is computationally too expensive for complex problems, metaheuristic algorithms are used to approximate such problems [23,24]. These algorithms do not necessarily find the global optimum, but they do find a good design.

One group of commonly used population-based metaheuristic algorithms is Genetic Algorithms (GA), which are inspired by natural evolution. Genetic Algorithms differ in the description of the

problem and in the implementation of the evolution process. Binary-coded GA represent variables as zero or one and are suitable for linear or nonlinear combinatorial problems [6,7,8,9,25]. Based on reported GA implementations, we implement a binary-coded Genetic Algorithm  $\text{GA}\left(\mathbf{d}, N_{\text{rec}}, P_{\text{mut}}, N_{\text{cov}}\right)$ , see Appendix A.3. This GA takes an initial population  $\mathbf{d}$ , a number of recombinations  $N_{\text{rec}}$  per generation, a mutation probability  $P_{\text{mut}}$  for each element per generation, and a number of generations without improvement  $N_{\text{cov}}$  for the convergence check.

The GA process consists out of selection-, recombination-, and mutation steps. The GA is initialized with a random population.

$$\mathbf{d} := \left( \text{random} \left( \mathbb{D} \left( \binom{N}{-el} \right) \right) \right)_{1 \leq i \leq N_{\text{pop}}} \quad (3)$$

with  $N_{\text{pop}}$  random designs. Each design is simulated and the fitness.

$$\text{fitness}(u) = \exp(-u) \quad (4)$$

is computed using the simulated endpoint displacement  $u$ . The exponential function guarantees a positive fitness that increases as  $u$  is minimized. In the selection step, designs are randomly sampled. The probability of sampling is proportional to the relative fitness of a design.

The recombination step finds  $N_{\text{rec}}$  random pairs out of the sampled designs and uses a block crossover operator as described by Kane [9]. The mutation step takes the designs from the combination step and computes a random real number between zero and one for each variable in a design. If the number is less than the mutation probability  $P_{\text{mut}}$ , the variable is flipped  $d_{ij} := 1 - d_{ij}$ .

In the mutation step, any design can be generated with a non-zero probability. This includes the globally optimal design. Consequently, GA is guaranteed to find the global optimum as the number of iterations approaches infinity. In reality, however, GA stops at some point and does not guarantee a globally optimal design.

The recombination and mutation step can produce invalid designs with free moving elements. A constraint handling method is required to handle invalid designs [26]. We use a repair method to make designs valid by removing free movable elements.

The designs are then passed on to the next generation which starts with the selection step again. We use a simple convergence check that defines a maximum number of generations without an improvement  $N_{\text{cov}}$ .

The parameters  $N_{\text{pop}}$ ,  $N_{\text{rec}}$ ,  $P_{\text{mut}}$  are optimized using Bayesian Optimization from the Python package scikit-optimize [27]. The goal is to minimize the relative costs that are defined in section 2.4. A Bayesian Optimization uses a stochastic surrogate model and tries to optimize the expected improvement [28]. This leads to a population size  $N_{\text{pop}} = 93$ , a number of recombinations per generation  $N_{\text{rec}} = 34$ , and a mutation probability  $P_{\text{mut}} = 5.5 \cdot 10^{-4}$ . The maximum number of iterations without any improvement for the convergence check is set to  $N_{\text{cov}} = 5$ . An alternative to the Bayesian Optimization would be an adaptive Genetic Algorithms as described by Balamurugan [6], where the parameters are adapted online during the optimization.



### 2.2.4. Simulated Annealing (SA)

Another metaheuristic algorithm is Simulated Annealing (SA) [10], which is inspired by the crystallization process during the annealing of metals. To work properly, SA must be able to generate each design in a finite number of steps. Then, SA converges to the global optimum, if the annealing process is slow enough [10,23]. In this work, however, a faster annealing process is used, that does not necessarily converge to the global optimum.

Like GA, SA is often used to approximately solve difficult combinatorial problems that cannot be solved exactly within a reasonable computational time [29]. Unlike GA, SA considers only one design at a time instead of maintaining an entire population. SA has already been used for structural optimization. Shim [11] optimizes a cantilever plate under tension and Jung [12] uses SA to, design of a resonator.

Our implementation of SA( $\mathbf{d}, T, \lambda_T, N_{it}$ ), see Appendix A.4, takes an initial design  $\mathbf{d}$ , an initial temperature  $T$ , a cooling rate  $\lambda_T$ , and a number of iterations  $N_{it}$ . SA modifies the current designs by flipping a random variable  $d_{ij} := 1 - d_{ij}$ . If the design becomes invalid due to freely movable elements, another variable is flipped instead. Next, the change  $\Delta u$  of the endpoint displacement  $u$  between the modified and the unmodified design is computed. If the endpoint displacement  $u$  improves and  $\Delta u \leq 0$ , the modified design is accepted. Otherwise, the probability of acceptance  $P_{\text{accept}}(\Delta u, T) = \exp(-\Delta u/T)$  is computed by the Boltzmann distribution. If the modified design is not an improvement, it is still accepted if a random number between zero and one is less than  $P_{\text{accept}}(\Delta u, T)$ . This allows SA to jump out of local optima and explore other designs.

The temperature  $T$  decreases by a cooling rate  $\lambda_T$  in each iteration, and the Boltzmann distribution becomes narrower and narrower. As a result, the acceptance probability decreases, and worse designs are rejected more often. SA continues to exploit the region near the current design more until it finds a better design and stops after a certain number of iterations  $N_{it}$ .

Grid search is used to find efficient optimization parameters. Therefore, fixed values are defined for each parameter and all combinations are computed. The optimized parameters are an initial temperature  $T = 0.1\text{mm}$ , a cooling rate  $\lambda_T = 0.999$ , and a number of iterations  $N_{it} = 50000$ . Adaptive Simulated annealing approaches [10] automatically adjust the cooling rate. Note that  $T$  is called temperature only for historical reasons, but it has the same unit as the endpoint displacement  $u$ .

### 2.2.5. Free Shape optimization (FS)

Shape optimization is a widely used approach to optimize geometries. It either optimizes parameters of given shape functions [16] or shifts nodes on an edge [17]. In contrast, our implementation of the Free Shape FS( $\mathbf{d}$ ) optimization, see Appendix A.5, takes an initial design  $\mathbf{d}$  as input and successively switches on and off elements lying on the edge of a corresponding design. Since FS is not able to generate all designs from the design space, FS cannot guarantee a globally optimal design.

Fig. 3 shows an iteration of FS. Green strokes indicate edge elements that can be added to the design, while orange strokes indicate removable elements. Starting from an initial design, FS modifies the design by swapping one element at a time. If the modified design is not an improvement, the element is reset to its original state and another edge element is considered. If the modified design is an improvement, FS accepts the modified design and starts modifying elements on the edge of the new design again. This is repeated until all elements on the edge are switched on or off and no improvement is found.

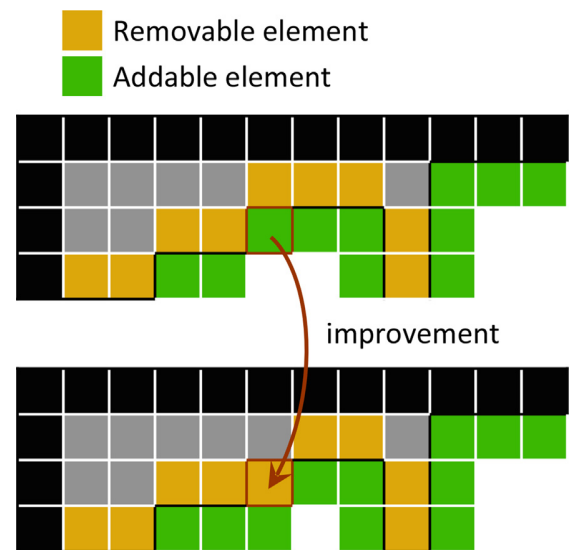


Fig. 3. Geometry modification options in Free Shape (FS) optimization.

## 2.3. Novel methods in topology optimization

### 2.3.1. Breadth-First search with Limited depth (BFSL)

The Breadth-First Search with Limited depth BFSL( $\mathbf{d}, \Delta d_{\text{max}}$ ), see Appendix A.6, takes an initial design  $\mathbf{d}$  and a maximum search depth  $\Delta d_{\text{max}}$  as input. Breadth-First Searches are used in a wide variety of combinatorial applications, ranging from finding minimal gene subsets for tumor classification [30] to task selection problems with limited resources for radar applications [31], but they are not yet used for topology optimization.

BFSL can be also be classified as a Reduced Variable Neighborhood Search (RVNS) [19]. Mladenović and Hansen [32] proposed Variable Neighborhood Search (VNS) algorithms that perform a local search. If no improvement is found, the search gets stuck in a local optimum. To escape, VNS searches for an improvement in a certain neighborhood and increases this neighborhood until a better solution is found or the termination condition is satisfied. RVNS skips the local search and relies entirely on the neighborhood search. Recent research on VNS methods has been done primarily in computer science [33], and VNS has already been used to optimize a truss structure [34].

BFSL systematically defines the neighborhood by a parameter  $\Delta d_{\text{max}}$  in a way for spatial problems. Fig. 4 shows how BFSL traverses designs in a stochastic breadth-first search with limited depth  $\Delta d_{\text{max}}$ . This means that BFSL first evaluates designs that are  $\Delta d = 1$  elements different from the current design. These designs are evaluated in a random order. If no improvement is found, BFSL evaluates designs that differ  $\Delta d = 2$  elements from the current design. This is repeated with increasing  $\Delta d$  up to the maximum depth  $\Delta d_{\text{max}}$ . However, as soon as a design is an improvement over the current design, the improved design becomes the new current design. Then, BFSL starts again to evaluate designs that differ by at most  $\Delta d_{\text{max}}$  elements have been computed and no improvement has been found. The number of such enumerated designs without an improvement

$$N_{\text{cov}}(\Delta d_{\text{max}}, N_{\text{var}}) \leq \sum_{i=1}^{\Delta d_{\text{max}}} \binom{N_{\text{var}}}{i} \quad (5)$$

depends on the maximum search depth  $\Delta d_{\text{max}}$  and the number of variables  $N_{\text{var}}$ . Note, that  $N_{\text{var}}$  is within the binomial coefficient and that  $N_{\text{cov}}$  increases rapidly with  $N_{\text{var}}$ .

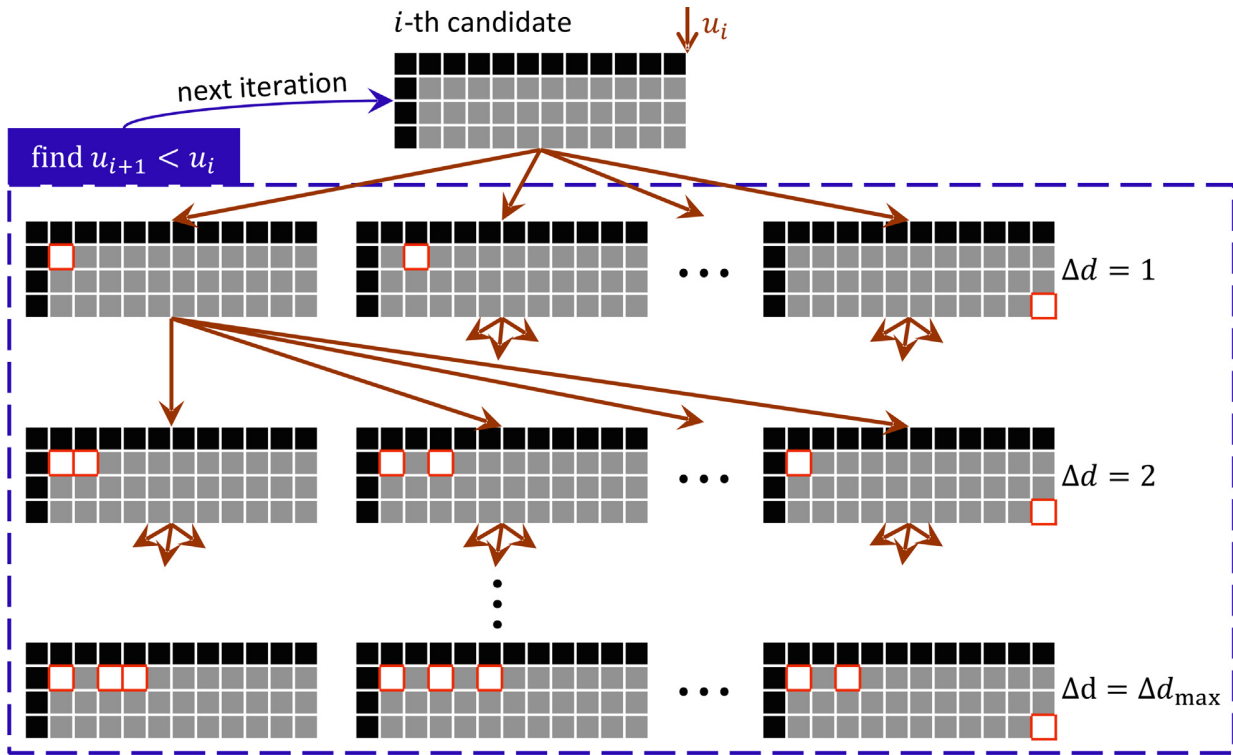


Fig. 4. Schematic of the Breadth-First Search with Limited depth (BFSL) algorithm. Designs that are at most  $\Delta d_{\max}$  different from the current design are explored until a better endpoint displacement  $u_{i+1}$  is found. This is repeated until no improvement can be found anymore.

If the maximum search depth is at least as large as the number of variables  $\Delta d_{\max} \geq N_{\text{var}}$ , every design from the design space is guaranteed to be generated. In this case, BFSL, like BF, guarantees a globally optimal solution. However, for problems whose local optima are also global optima  $\Delta d_{\max} = 1$  is sufficient to find the optimum. For realistic problems, the smallest  $\Delta d_{\max}$  necessary to find the global optimum is somewhere between 1 and  $N_{\text{var}}$ .

2.3.2. Truncated Breadth-First search with Limited depth (TBFSL)

In addition to BFSL, the Truncated Breadth-First Search with Limited depth TBFSL( $\mathbf{d}, \Delta d_{\max}, \Delta r_{\max}$ ), see Appendix A.7, allows only simultaneously modified elements that are at most  $\Delta r_{\max}$  elements apart. As can be seen in Fig. 5, the distance  $\Delta r$  between different elements is computed using the  $l_1$  norm. Like BFSL, TBFSL is not yet used for topology optimization.

The additional distance restriction reduces the required number of iterations without any improvement.

$$N_{\text{cov}}(\Delta d_{\max}, \Delta r_{\max}, N_{\text{var}}) \leq N_{\text{var}} \cdot \sum_{i=1}^{\Delta d_{\max}} \left( 2 \cdot \binom{\Delta r_{\max} + 0.5}{i} - 1 \right) \tag{6}$$

before TBFSL stops. Unlike BFSL, the number of iterations depends only linearly on  $N_{\text{var}}$ .

To guarantee a globally optimal design, the maximum search depth should be to be at least as large as the number of variables  $\Delta d_{\max} \geq N_{\text{var}}$ , and the maximum distance  $\Delta r_{\max}$  should be at least as large as the distance between the two most distant variable elements.

2.3.3. Multi-Grid optimization (MG)

To find a good optimum, a stepwise refinement of the grid is performed in the following procedures. The idea of solving a problem on grids with increasing refinement is not new: It is already

used for multi-grid solvers [35,36] and is also applied to very different problems such as image processing with multiscale operators [37,38], prediction of local stress fields [39], or decoupling of variables from the FEM mesh [40]. However, automatic stepwise refinement has not yet been applied to topology optimization.

The Multi-Grid MG( $\mathbf{d}, (N_{-1}, \mathcal{A}_1), (N_{-2}, \mathcal{A}_2), \dots$ ) optimization, see Appendix A.8, takes as input an initial design  $\mathbf{d}$  and a sequence of mesh sizes  $N_i$  and corresponding optimization algorithms  $\mathcal{A}_i$ . Fig. 6 shows the mapping process from a coarse mesh to a finer mesh. MG starts with an initial design  $\mathbf{d}$ . First, a coarse voxel mesh is optimized using one of the previously mentioned algorithms. When the optimizer has converged using the coarse mesh, a finer voxel mesh is generated, and the geometry is mapped onto this mesh. Each new design element looks for the nearest element in the coarse mesh and takes its value. The finer mesh is then optimized based on the mapped design. This process is repeated until the desired grid size is reached.

Since a voxel in the coarse mesh covers a larger region, turning an element on or off will result in a completely different design. Consequently, the coarse mesh optimization explores a wide variety of designs. The fine mesh optimization, on the other hand, turns on or off smaller elements in the finer mesh and exploits similar-looking designs.

Whether MG guarantees a globally optimal design depends only on the algorithm run on the last mesh. If that algorithm converges to the global optimum, then so does MG.

2.4. Efficiency benchmark

When all valid designs  $\mathbf{d}$  in a design space  $\mathbb{D}$  are computed using BF, the rank  $k$  of each design can be computed according to the endpoint displacement  $u$ . The best design has rank  $k = 1$  and

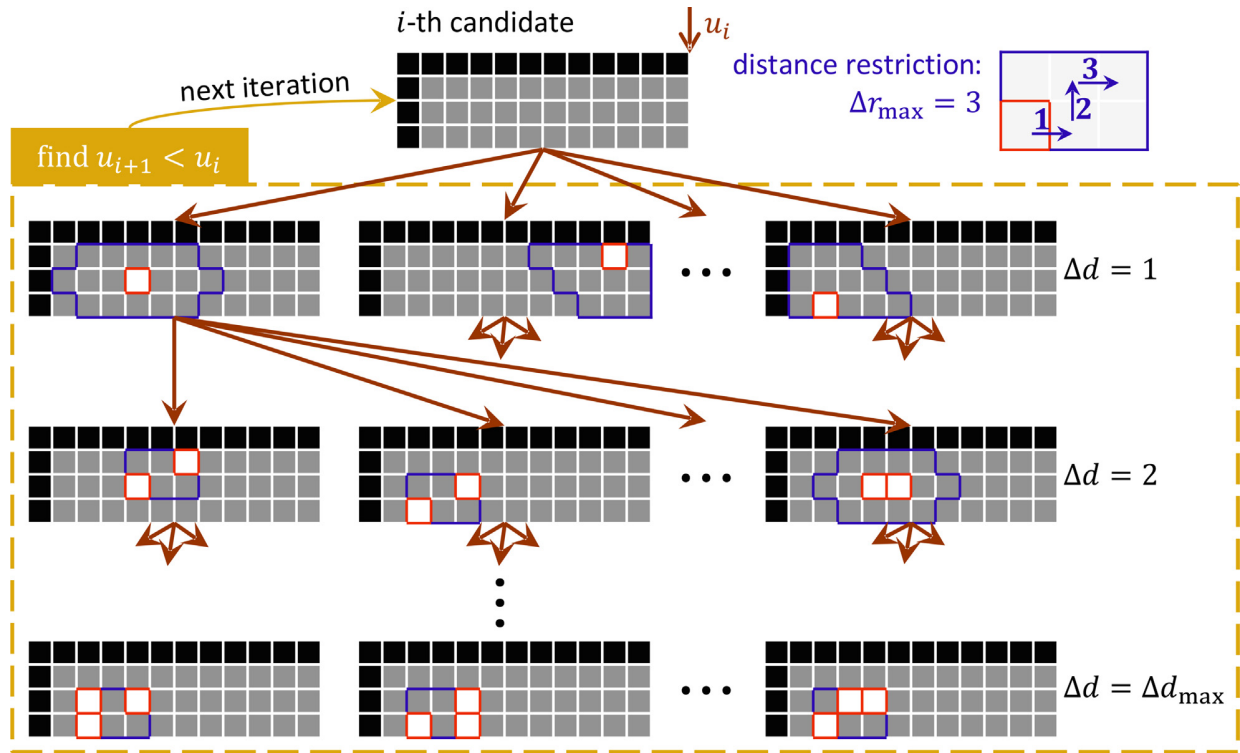


Fig. 5. Schematic of the Truncated Breadth-First Search with Limited depth (TBFSL). Unlike the BFSL approach, an additional distance restriction forces elements that have changed from the previous design to be at most  $\Delta r_{\max}$  elements away from each other.

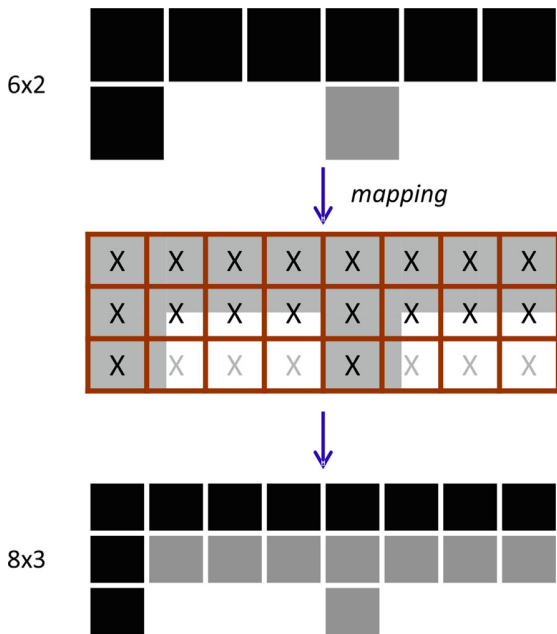


Fig. 6. The Multi-Grid (MG) approach maps an optimized  $6 \times 2$  design to an  $8 \times 3$  design that can be further optimized.

the rank of the worst design is equal to the number of valid designs in the design space  $k = |\mathbb{D}|$ .

The rank is used to compare the optimization algorithms. All optimization algorithms require a different number of simulations  $N_{\text{sim}}$  and find designs whose rank  $k$  varies. If one method finds a better design with a lower rank, but the other method is faster with fewer simulations  $N_{\text{sim}}$ , it is not clear which method is the

better one. Therefore, a performance measure is needed for comparison that considers both, the rank of the design  $k$  and the number of simulations  $N_{\text{sim}}$ . The idea of our performance measure is to compare the number of simulations of an optimization method  $N_{\text{sim}}$ , with the number of simulations RS needs on average to find an equal or better design with rank  $\leq k$ ,  $N_{\text{sim}}^{\text{RS}}$ . The latter can be calculated as

$$N_{\text{sim}}^{\text{RS}}(k) = \sum_{i=1}^{|\mathbb{D}|} i \cdot P(i|k) = 1 + \frac{|\mathbb{D}| - k}{1 + k} \quad (7)$$

with the number of designs  $|\mathbb{D}|$  in the design space  $\mathbb{D}$ . The average number of simulations of RS  $N_{\text{sim}}^{\text{RS}}$  is computed by weighting the iterations  $i$  by their probability under the condition of finding a design with rank  $\leq k$ . This probability

$$P(i|k) = \underbrace{\frac{k}{|\mathbb{D}| - i + 1}}_{\text{find in } i\text{-th iteration}} \cdot \underbrace{\prod_{j=1}^{i-1} \left(1 - \frac{k}{|\mathbb{D}| - j + 1}\right)}_{\text{do not find in previous iterations}} = \text{NHG}(i - 1 \mid |\mathbb{D}|, |\mathbb{D}| - k, 1) \quad (8)$$

is a special case of the negative hypergeometric distribution NHG as described by Johnson [41]. This distribution considers a population of size  $|\mathbb{D}|$  in which  $|\mathbb{D}| - k$  designs have rank  $> k$ . From this population, unique designs are sampled until a design with rank  $\leq k$  is selected. Then the sampled designs are counted, excluding the last one. The count is equal to  $i - 1$ .

Now, the performance of an optimization algorithm that returns the  $k$ -th best design after  $N_{\text{sim}}$  simulations can be measured in terms of the relative computational cost.

$$c(N_{\text{sim}}, k) = \frac{N_{\text{sim}}}{N_{\text{sim}}^{\text{RS}}(k)} \quad (9)$$

The smaller the relative cost, the more efficient the optimization algorithm is.

### 3. Results and discussion

The cantilever beam defined in section 2.1 is optimized using the previously defined optimization methods. Therefore, all possible designs are computed on a coarse grid according to the BF method and the optimization methods are compared with each other. Then, the cantilever beam is optimized on a finer grid using a multi-grid approach.

For the comparison of the multi-grid optimization, the methods are evaluated in terms of computational time, which was obtained using a desktop computer with an Intel i5-6500 CPU with 3.2 GHz, four logical and physical cores, and 16 GB RAM. Since solving the FEM system is only part of the code written, the elapsed real time is used as the computation time. One optimization simulates up to four designs in parallel.

#### 3.1. Brute-Force search on a coarse grid

In this section, all designs  $\mathbf{d} \in \mathbb{D}((8, 5))$  are computed using BF. Since the first row and column elements are fixed, there are  $7 \times 4 = 28$  binary variables. Thus,  $2^{28} = 268\,435\,456$  element combinations are possible. However, most of these combinations contain freely movable elements and are therefore invalid. Only 59554032 designs are valid. The computation of all valid designs takes about 10 hours. Therefore, the effective average computation time for a simulation is 0.56 ms, but since we run four simulations in parallel, the real average computation time for a simulation is 2.24 ms.

Fig. 7 plots the endpoint displacement  $u$  and the number of elements for each design in a two-dimensional histogram. The color of each box corresponds to the number of valid designs in that region. Note the logarithmic scale of the color bar. Almost all designs have between 20 and 35 elements and an endpoint displacement  $u$  between 0.0 mm and 0.1 mm. The computation of all valid designs  $\mathbf{d} \in \mathbb{D}((8, 5))$  allows to rank the designs according to their endpoint displacement  $u$ .

The geometry of four designs, labeled a–d in Fig. 7, is shown in Fig. 8. The best design with rank  $k$ , shown in Fig. 8a, has the lowest endpoint displacement, which is actually negative, so the beam moves up at the end. In contrast, the worst design in Fig. 8b has

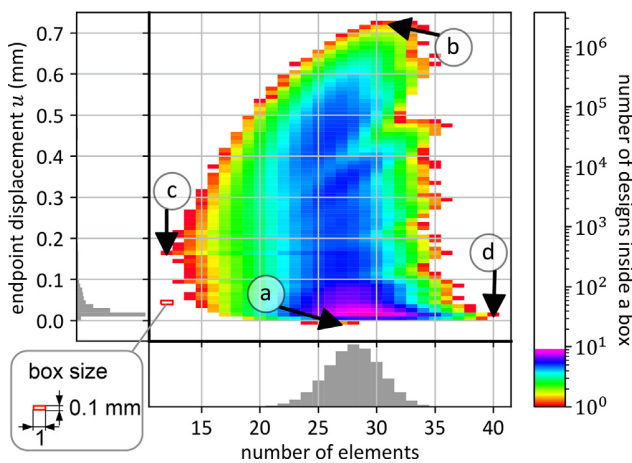


Fig. 7. Results of the BF computation of all valid designs  $\mathbf{d} \in \mathbb{D}((8, 5))$ . The designs are grouped by the number of elements on the  $x$ -axis and the endpoint displacement  $u$  on the  $y$ -axis. The color of each box indicates the number of designs contained in that box. The designs labeled a–d are shown in Fig. 8.

a large mass hanging at the end that pulls the endpoint down. The lightest design in Fig. 8c contains only the 12 fixed elements in the left column and the top row. The heaviest design in Fig. 8d contains 40 elements. Note that the deformations drawn are scaled by factors between 10 and 1000.

#### 3.2. Benchmark on a coarse grid

The results of the Brute-Force search on an  $8 \times 5$  grid, presented in the previous section 3.1, are used to compare the optimization algorithms to each other. Therefore, the ranks  $k$  and relative costs  $c$  are computed as described in section 2.4.

Table 2 lists the resulting ranks and relative costs of all optimization algorithms. The algorithms are ranked according to their efficiency. Since most algorithms are stochastic and the result of a single run is not representative, 100 runs with randomly chosen initial designs  $\mathbf{d} = \text{random}(\mathbb{D})$  are performed for each algorithm. Each cell in the table shows the observed 25%, 50% (median), and 75% percentiles of the 100 results obtained. In addition, the probability density distributions of the relative costs  $c$  are plotted in the last column of Table 2. The probability density distributions are estimated using the kernel density estimator of scipy [42], which uses Scott's rule [43].

BF always finds the best design with rank  $k = 1$  as shown in Fig. 8a. However, BF is the most inefficient algorithm with a relative cost of  $c = 200\%$  because it simulates every valid design.

As can be seen in Table 2, RS is on average about twice as efficient as BF, since it finds the best design on average after simulating half of the designs. If a much higher number of repetitions of RS were performed, the relative cost  $c_{50\%} = 79.496\%$  would approach the theoretically correct average relative cost of 100%.

SA has about half of the median cost  $c_{50\%} = 52.265\%$  of RS. Fig. 9 shows a typical design optimized with SA. After BF and RS, SA is the least efficient optimization method for this problem. After 10000 iterations, the temperature drops to  $T = 0.1 \text{ mm} \cdot 0.999^{10000} \approx 4.5 \cdot 10^{-6} \text{ mm}$  and the probability of accepting worse designs is negligible. Accepting worse designs would be necessary to escape a local minimum and explore further regions. Exploring further regions is crucial to achieving a good design but is not very likely after 10,000 iterations. SA mainly exploits the actual local minimum for the remaining 40,000 iterations.

The next best approach is BFSL( $\mathbf{d}, 4$ ) with a search depth  $\Delta d_{\max} = 4$ . Although the initial design  $\mathbf{d}$  is chosen randomly, BFSL( $\mathbf{d}, 4$ ) most often finds the same design with rank  $k = 202$  as shown in Fig. 9b. This is the best design found by any of the optimization algorithms except BF, while BFSL( $\mathbf{d}, 4$ ) requires on average  $N_{\text{sim},50\%} = 13356$  simulations and causes only  $c_{50\%} \approx 4.5\%$  of the cost of RS.

FS is the algorithm that requires the fewest simulations with  $N_{\text{sim},50\%} = 47$ . However, the optimized designs have high ranks. Fig. 9c shows an optimized design with cavities. FS does not vary elements within the design and cavities appear only when the initial design  $\mathbf{d}$  contains the cavities. Consequently, the optimized design is highly dependent on the initial design, and since the initial design is randomly chosen, the relative costs vary over a wide range with a 25% percentile of  $c_{25\%} \approx 0.4\%$  and a 75% percentile of  $c_{75\%} \approx 8.0\%$ .

GA provides both a low median rank  $k_{50\%} = 508$  and a low number of simulations  $N_{\text{sim},50\%} = 967$ . This results in an average relative cost of  $c_{50\%} \approx 0.873\%$ . Fig. 9d shows a design optimized by GA.

TBFSL( $\mathbf{d}, 4, 2$ ) is an extension of BFSL( $\mathbf{d}, 4$ ) that restricts the modified elements to be at most  $\Delta r_{\max} = 2$  elements away from each other. This additional restriction significantly reduces the number of simulations from about 13356 to about 330. At the same time, the median rank increases only slightly from 202 to

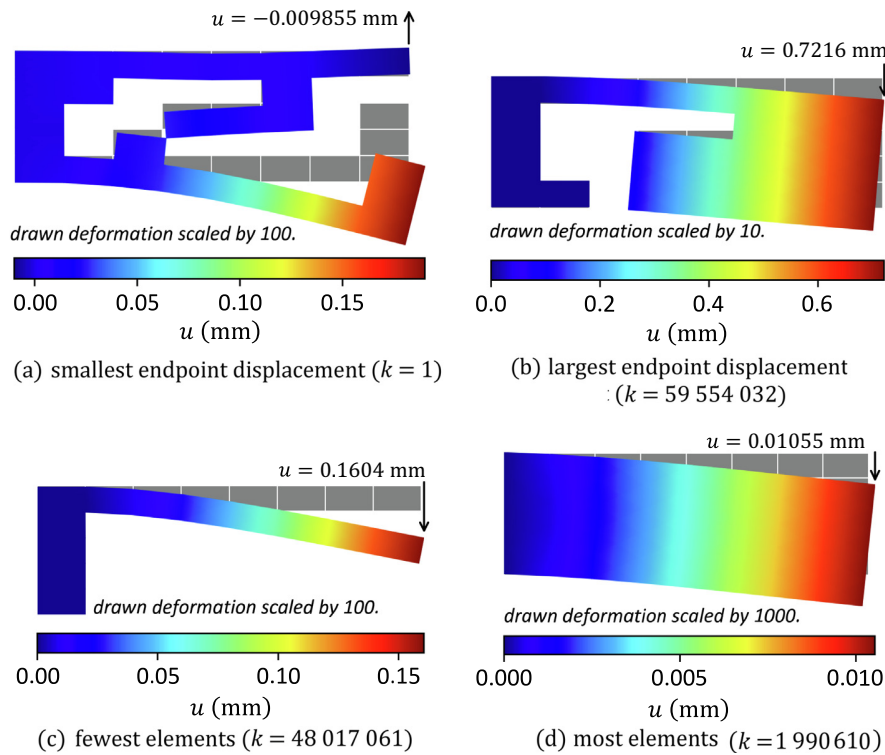


Fig. 8. Specific designs shown in Fig. 7.

247. Fig. 9e shows the result with rank  $k = 247$  and the dashed contour illustrates the design with rank  $k = 202$ . Five elements are different between the two designs and the different elements lie in a rectangular area of size  $3 \times 4$ . This corresponds to an  $l_1$ -norm of  $|3| + |4| = 7$ . Consequently, TBFSL would be able to go directly from the 247th best design to the 202nd best design if the search depth was set to  $\Delta d_{\max} = 5$  and the search radius was increased to  $\Delta r_{\max} = 7$ . However, this would significantly increase the number of simulations and TBFSL would lose its performance advantage over BFSL.

The most efficient methods are BFSL( $\mathbf{d}, 1$ ) and TBFSL( $\mathbf{d}, 1, 0$ ) with a search depth  $\Delta d_{\max} = 1$ . Fig. 9f shows a design optimized by BFSL( $\mathbf{d}, 1$ ) and TBFSL( $\mathbf{d}, 1, 0$ ). Actually, BFSL( $\mathbf{d}, 1$ ) and TBFSL( $\mathbf{d}, 1, 0$ ) are identical because the additional constraint of TBFSL has no effect when only one element is modified at a time. However, Table 2 shows different results for the two methods. The reason for this is that with a search depth of  $\Delta d_{\max} = 1$ , the optimizer falls into the very first local optimum, and the result thus depends strongly on the random initial design.

### 3.3. Parameter study for TBFSL

Fig. 10 shows the effect of the TBFSL parameters, the maximum search depth  $\Delta d_{\max}$  and the maximum size of the modified region  $\Delta r_{\max}$ , on the relative costs and the ranks of the optimized designs. Each parameter setting is run 100 times to compute the median relative cost  $c_{50\%}$  and the rank  $k_{50\%}$ . The interquartile range (IQR) is a measure of scatter and is calculated as the difference between the 75% and the 25% percentiles.

The IQR is shown in Fig. 10a. As explained in the previous section, all methods with a search depth of  $\Delta d_{\max} = 1$  (blue) are identical, but scatter due to the dependence on the random initial design. Since the maximum distance between two variable elements on this  $8 \times 5$  grid is 9, the additional restriction of TBFSL

has no effect for settings with  $\Delta r_{\max} = 9$ , so that TBFSL( $\mathbf{d}, \Delta d_{\max}, 9$ ) is identical to BFSL( $\mathbf{d}, \Delta d_{\max}$ ).

All investigated parameter settings have costs  $< 10\%$  of the RS approach, which is highlighted by a horizontal red line in Fig. 10a. The relative costs on the logarithmic scale increase significantly with the search depth  $\Delta d_{\max}$ . As the distance  $\Delta r_{\max}$  increases, the relative costs increase until they plateau at  $\Delta r_{\max} \approx 6$ .

Although running TBFSL or BFSL with a search depth  $\Delta d_{\max} = 1$  is the most efficient approach, the ranks of the optimized designs are higher. Except for  $\Delta d_{\max} = 1$ , all settings result in optimized designs with ranks  $< 300$ . When the size of the modified area is  $\Delta r_{\max} \geq 4$ , the ranks  $k_{50\%}$  decrease slightly.

### 3.4. Multi-grid optimization

This section compares the performance of the multi-grid (MG) optimization with that of a single-grid optimization. Therefore, both approaches optimize the design using a final grid of size  $48 \times 16$ . This grid contains  $47 \times 15 = 705$  variables compared to  $7 \times 4 = 28$  variables of the previous  $8 \times 5$  grid. It is no longer possible to perform a brute-force optimization on the finer grid because the number of designs is on the order of  $2^{705} \approx 10^{212}$ . Consequently, the rank of an optimized design cannot be computed and RS cannot be used as a benchmark method as in the previous section. Therefore, the optimization methods are thus compared directly by considering the optimized endpoint displacement  $u$  as well as the computation time  $t$ . Since TBFSL is the most efficient optimization algorithm for the  $8 \times 5$  grid, it is used within the multi-grid and single-grid approaches.

Table 3 shows the results of the single-grid approach for the  $48 \times 16$  grid. We use a search depth of  $\Delta d_{\max} = 4$  and limit the size of the modified area to  $\Delta r_{\max} = 2$ . Due to the long computation time of up to 6 : 42h for one optimization, only 11 runs are per-

**Table 2**  
Results of the used optimization algorithms. Except for BF, each algorithm is repeated 100 times. BF is computed once. The figures in the last column show an estimated probability density distribution of  $c$ . The black point marks the median. The red line indicates the average number of simulations RS would need to get the same or a better design.

Optimization algorithm	Percentile	Endpoint displ. $u$ (mm)	Rank $k(1)$	Simulations $N_{sim}(1)$	Relative costs	
					$c(\%)$	
BF( $N_{-el}$ )		-0.00986	1	59554032	200.00	
RS( $N_{-el}, -0.00986\text{mm}$ )	25%	-0.00986	1	13216395	44.385	
	50%	-0.00986	1	23671511	79.496	
	75%	-0.00986	1	43164841	144.960	
SA( $\mathbf{d}, 0.1\text{mm}, 0.999, 50000$ )	25%	0.00589	267	50001	22.480	
	50%	0.00621	622	50001	52.265	
	75%	0.00641	1182	50001	99.345	
BFSL( $\mathbf{d}, 4$ )	25%	0.00562	202	11411	3.862	
	50%	0.00562	202	13356	4.474	
	75%	0.00562	202	16331	5.567	
FS( $\mathbf{d}$ )	25%	0.00680	4186	36	0.401	
	50%	0.00751	28810	47	2.319	
	75%	0.00823	122505	58	8.062	
GA( $(\mathbf{d})_{1 \leq i \leq 93}, 34, 5.5 \cdot 10^{-4}, 5$ )	25%	0.00595	296	813	0.566	
	50%	0.00623	508	967	0.875	
	75%	0.00643	1290	1114	1.896	
TBFSL( $\mathbf{d}, 4, 2$ )	25%	0.00562	202	264	0.103	
	50%	0.00582	247	330	0.126	
	75%	0.00582	247	391	0.149	
BFSL( $\mathbf{d}, 1$ )	25%	0.00582	247	81	0.044	
	50%	0.00632	875	101	0.099	
	75%	0.00671	3224	116	0.551	
TBFSL( $\mathbf{d}, 1, 0$ )	25%	0.00582	247	86	0.045	
	50%	0.00593	299	105	0.065	
	75%	0.00667	2851	126	0.304	

formed instead of the 100 runs in the previous section. The number of simulations  $N_{sim}$ , the computation time  $t$ , and the optimized endpoint displacement  $u$  for each of these runs are sorted by the endpoint displacement. Runs that result in the median endpoint displacement for the single-grid and multi-grid optimizations are treated as representative runs. Compared to the coarse designs of the brute force optimization, much better designs in terms of endpoint displacement can be found for this finer grid.

Table 4 lists the results of the 11 MG runs, sorted by the endpoint displacement from the best run to the worst run. Like the one-grid approach, MG starts with a random initial design  $\mathbf{d}$ . Unlike the one-grid approach, MG uses 9 grids of increasing number of cells. For each grid, the grid size and an optimization algorithm  $\mathcal{A}$  are passed to MG. Since the simulation on a coarse grid is faster than that on a fine grid, higher values  $\Delta d_{max}$  and  $\Delta r_{max}$  can be chosen on the coarse grid. As defined in Table 4, MG first optimizes the initial design on a  $10 \times 10$  grid using the algorithm  $\mathcal{A}_1$ , which performs a TBFSL optimization with  $\Delta d_{max} = 4$  and  $\Delta r_{max} = 6$ . The resulting design is then optimized on a  $12 \times 12$  grid using the optimization algorithm  $\mathcal{A}_2$ . This continues until the desired grid size of  $48 \times 16$  is reached.

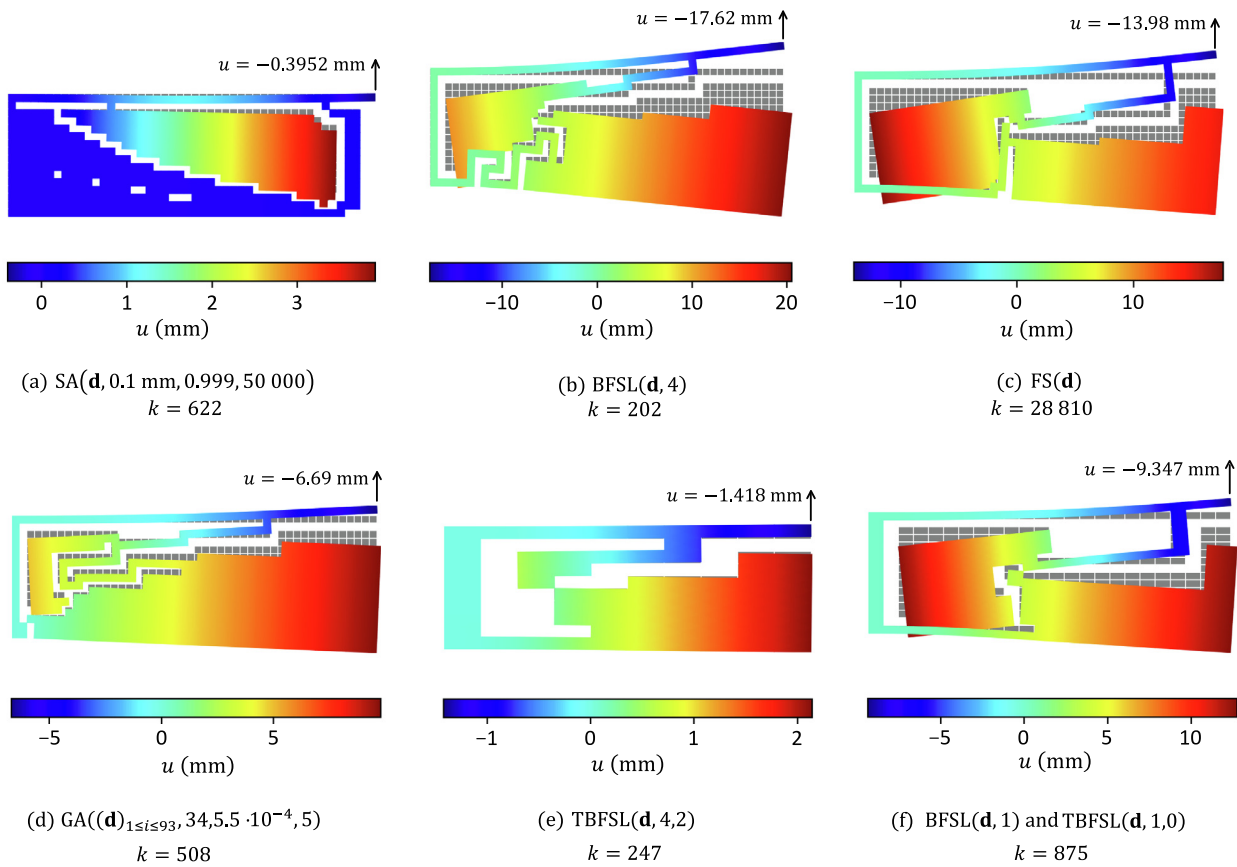
Compared to the one-grid approach in Table 3, MG takes a median of about half the time of the single-grid approach and provides a better median of the endpoint displacement. Furthermore, the computational time of MG scatters less, as can be seen by from the IQR, and is therefore more predictable.

Fig. 11a shows this process of increasing the grid size. The endpoint displacement is plotted over the total computation time and

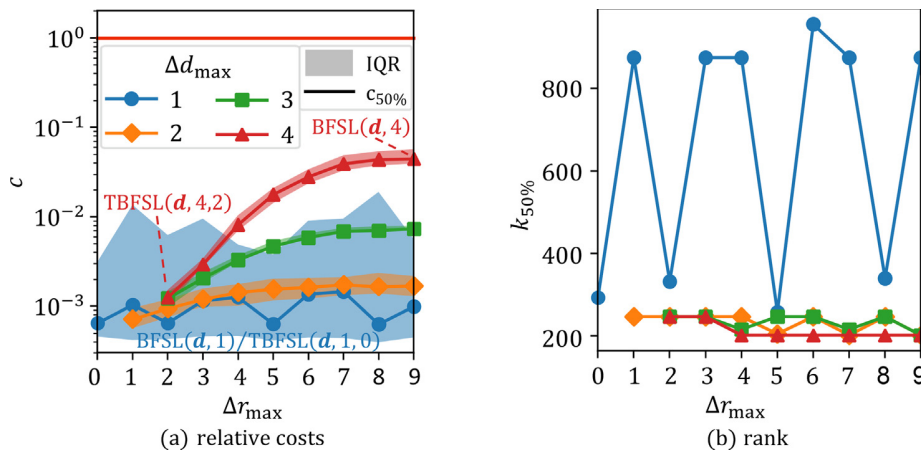
the line colors correspond to the grid sizes and TBFSL parameters as shown in the legend.

Most grid sizes are not multiples of the previous grid size and therefore do not match exactly. If a design is scaled to the next grid that does not exactly match the previous design, the design is scaled imprecisely and the endpoint displacement may be worse. This effect can be seen in Fig. 11a at the transition from the brown  $20 \times 14$  grid to the pink  $22 \times 15$  grid. However, the multi-grid optimization quickly recovers the previous endpoint displacement.

A drawback of MG is the higher number of parameters introduced. The size of each grid has to be selected, as well as the TBFSL parameters  $\Delta d_{max}$  and  $\Delta r_{max}$ . In this example, 36 parameters have to be specified before the optimization starts. Algorithms such as Bayesian Optimization are designed to find such parameters that have to be known in advance. However, methods using interpolation schemes or surrogate models like Bayesian Optimization are not suitable for such high-dimensional problems, because they have to evaluate the multi-grid approach with an exponential number of parameter combinations [28]. Therefore, the authors manually selected the optimization parameters based on the observations made in the previous section. Since the  $8 \times 5$  design from above is already limited in the possible variants, a starting mesh with a grid size of  $10 \times 10$  is used. The parameters  $\Delta d_{max}$  and  $\Delta r_{max}$  of the used TBFSL are derived from the parameter evaluations in Fig. 10 for the initial design. It is assumed that the designs vary less each time the mesh is refined. Therefore, the parameter  $\Delta d_{max}$  is reduced to 2 for the final mesh. This speeds up the computation significantly, since the computational time



**Fig. 9.** Examples of optimized designs for each of the optimization algorithms listed in Table 2. The plotted deformation is scaled by a factor of 100. TBFSL(d, 4, 2) adds an additional constraint in addition to BFSL(d, 4) and therefore returns on average a slightly worse design (e) compared to (b). The dashed line shows this difference.



**Fig. 10.** Effect of the search depth  $\Delta d_{\max}$  and modified region size  $\Delta r_{\max}$  on the median relative cost  $c_{50\%}$ , the interquartile range (IQR), and the median rank  $k_{50\%}$ . The horizontal red line in (a) marks the relative cost of RS. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

for one iteration significantly increases with the number of elements. A simulation on a  $10 \times 10$  grid is 10 to 50 times faster than a simulation on a  $48 \times 16$  grid.

Fig. 11b shows the convergence curves of the single-grid (blue) and multi-grid (red) runs. The representative runs (those with the median endpoint displacements out of a sample of 11) are highlighted and marked with c and d. Fig. 11c shows the final endpoint displacements and the computational time of all runs. As can be seen in Fig. 11c, most of the multi-grid runs (red points) require

less computation time than the single-grid runs (blue points), although the multi-grid approach at the median requires about three times more simulations. Furthermore, the multi-grid runs tend to have a smaller endpoint displacement and return a good design quite reliably within a reasonable time, while the one-grid results show more scatter.

However, the worst design a is achieved by the multi-grid approach which is shown in Fig. 12a. Since only one of the multi-grid designs has such a large endpoint displacement, this run can

**Table 3**  
Results of 11 runs performed on a  $48 \times 16$  grid using TBFSL with a search depth  $\Delta d_{\max} = 4$  and a of the modified area size  $\Delta r_{\max} = 2$ . The initial design  $\mathbf{d}' := \text{random}(\mathbb{D}(48, 16))$  is chosen randomly for each run. The runs are sorted by the endpoint displacement.

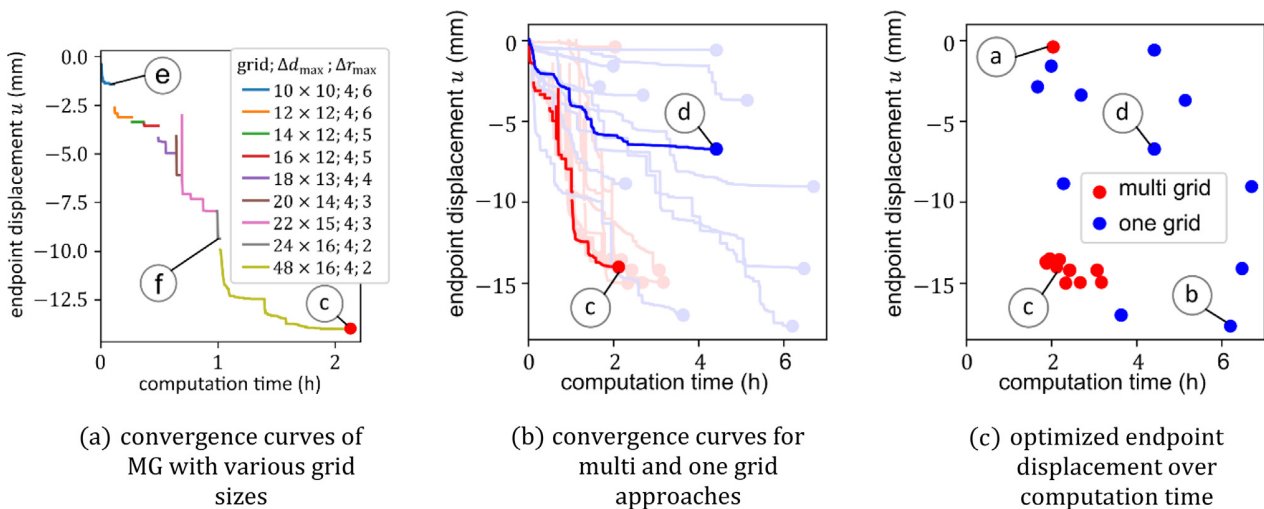
Optimization algorithm	Run	Simulations $N_{\text{sim}}(1)$	Computation time $t(\text{s})$	Endpoint displ. $u(\text{mm})$
TBFSL( $\mathbf{d}, 4, 2$ )	1	364675	22312	-17.62
	2	222134	13047	-16.95
	3	359095	23296	-14.08
	4	416722	24102	-9.01
	5	131115	8174	-8.84
	6*	243381	15869	-6.69
	7	259145	18483	-3.68
	8	144951	9666	-3.37
	9	157372	5972	-2.85
	10	161641	7132	-1.57
	11	193293	15868	-0.58
	Median	222134	15868	-6.69
	IQR	149614	11478	8.44

1\* representative run

**Table 4**  
Results of 11 runs performed on nine grids starting from a coarse  $10 \times 10$  grid and going up to a fine  $48 \times 16$  grid. Each grid is optimized using TBFSL with a search depth  $\Delta d_{\max} = 4$  and a decreasing size of the modification range  $\Delta r_{\max}$ . The initial design  $\mathbf{d}' := \text{random}(\mathbb{D}(10, 10))$  is chosen randomly for each run. The runs are sorted by the endpoint displacement.

Optimization algorithm	run	Simulations $N_{\text{sim}}(1)$	Computation time $t(\text{s})$	Endpoint ispl. $u(\text{mm})$
MG( $\mathbf{d}'$ , ((10, 10), $\mathcal{A}_1 : \mathbf{d} \rightarrow \text{TBFSL}(\mathbf{d}, 4, 6)$ ), ((12, 12), $\mathcal{A}_2 : \mathbf{d} \rightarrow \text{TBFSL}(\mathbf{d}, 4, 6)$ ), ((14, 12), $\mathcal{A}_3 : \mathbf{d} \rightarrow \text{TBFSL}(\mathbf{d}, 4, 5)$ ), ((16, 12), $\mathcal{A}_4 : \mathbf{d} \rightarrow \text{TBFSL}(\mathbf{d}, 4, 5)$ ), ((18, 13), $\mathcal{A}_5 : \mathbf{d} \rightarrow \text{TBFSL}(\mathbf{d}, 4, 4)$ ), ((20, 14), $\mathcal{A}_6 : \mathbf{d} \rightarrow \text{TBFSL}(\mathbf{d}, 4, 3)$ ), ((22, 15), $\mathcal{A}_7 : \mathbf{d} \rightarrow \text{TBFSL}(\mathbf{d}, 4, 3)$ ), ((24, 16), $\mathcal{A}_8 : \mathbf{d} \rightarrow \text{TBFSL}(\mathbf{d}, 4, 2)$ ), ((48, 16), $\mathcal{A}_9 : \mathbf{d} \rightarrow \text{TBFSL}(\mathbf{d}, 4, 2)$ ))	1	927706	8358	-14.98
	2	731563	9599	-14.94
	3	1037718	11385	-14.92
	4	826378	8693	-14.18
	5	733592	11022	-14.18
	6*	700784	7591	-13.98
	7	658132	6741	-13.75
	8	588992	6681	-13.67
	9	588461	7813	-13.52
	10	648319	7009	-13.48
	11	707768	7286	-0.40
	Median	707768	7813	-13.98
	IQR	126760	1999	0.96

1\* representative run



**Fig. 11.** Comparison of the multi-grid (red) and the single-grid (blue) approach. For one multi-grid optimization, (a) plots the endpoint displacement over the computation time for the varying TBFSL parameters  $\Delta d_{\max}$  and  $\Delta r_{\max}$ . All eleven cases of the single-grid and the multi-grid approaches are plotted in (b), and the optimized endpoint displacement  $u$  and the computation time for all runs of each approach are shown in (c). The annotated designs a-f are plotted in Fig. 12. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



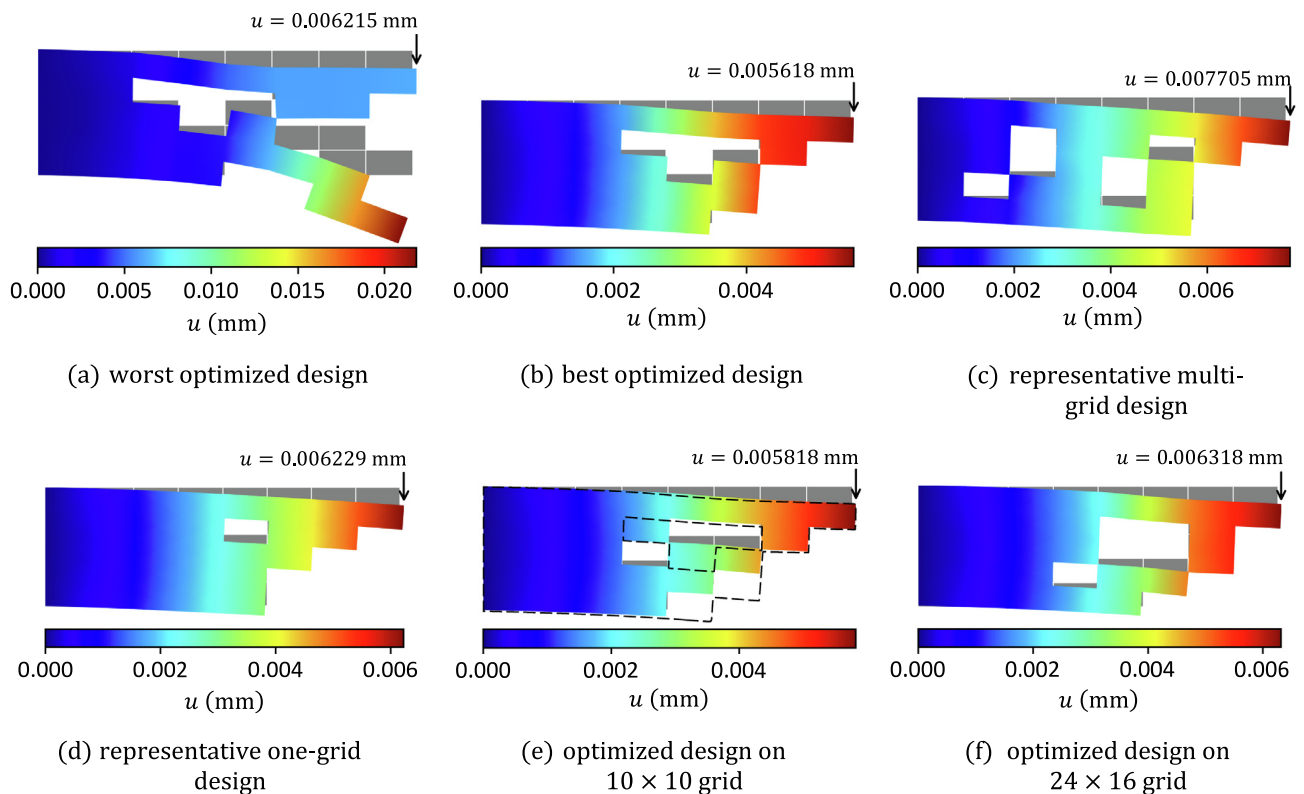


Fig. 12. Designs shown in Fig. 11 as a–f. The drawn deformation is scaled by a factor of 0.5. Self-contact and large deformations are not considered.

be considered a statistical outlier. The three designs labeled e, f, and c in Fig. 11a are snapshots of the representative multi-grid run after the first, next-to-last, and last refinement. They are shown in Fig. 12e, f, and c. The basic principle of the final design c is already present in the coarse design e after a few minutes. After about an hour, the design f is obtained on the  $24 \times 16$  grid. This design already looks very similar to the final design. The remaining hour is used to optimize the design on the finest grid.

The best design b is obtained using the single-grid method and is shown in Fig. 12b. Designs a and c obtained by the multi-grid approach appear less complicated than the designs b and d of the single-grid approach. One reason could be that complicated features are destroyed during the imprecise scaling to the next grid. As a result, the multi-grid approach yields a design that is less grid dependent than the single-grid approach.

#### 4. Conclusions

This work evaluates a number of optimization methods for highly nonlinear structural problems is evaluated. As an example problem, a discrete 2D plane stress cantilever beam under dead-weight load is optimized using these methods. This problem is found to be both simple and highly nonlinear. Commonly used methods such as Brute-Force search, Random Sampling, Simulated Annealing, Genetic Algorithm, and Free Shape optimization are compared with our methods BFSL, TBFSL which are based upon a Reduced Variable Neighborhood Search and a Breadth-First Search. These methods have not yet been applied to topology optimization.

Each optimization method differs in the quality of the result as well as in the number of simulations required. For a comparison, both quality and number of simulations need to be considered. Therefore, an efficiency measure is introduced that indicates how

many simulations are required compared to how many simulations would be required by Random Sampling to archive the same or better quality.

The Brute Force search always finds the global optimum, but it is twice as inefficient as the Random Sampling. In contrast, Simulated Annealing is twice as efficient, the Free Shape optimization is about 43-times more efficient, the Genetic Algorithm is about 114-times more efficient, and BFSL and TBFSL are up to 1540-times more efficient than the Random Sampling. However, except for the Brute Force search, none of the optimization methods finds the global optimum.

Since TBFSL is most efficient, it is also used to optimize a finer mesh grid. Therefore, TBFSL is applied once directly to the fine mesh grid. Alternatively, TBFSL is applied successively to multiple grids that become finer and finer until the desired mesh size is reached. It turns out that this multi-grid approach finds, on average, a better beam design in about half the computational time compared to the direct approach. The settings of the multi-grid optimization were chosen manually in this work and could be optimized in future work.

Note that the optimization algorithms studied rely on a model that computes fast enough to run thousands of iterations in a reasonable time frame. For realistic applications of finite element models, this may not be the case.

Furthermore, since the resolution of the FEM mesh is coupled to the representation of the variables, a finer FEM mesh significantly increases the number of variables and the optimization time. To overcome this problem, the FEM mesh can be decoupled from the variable representation either by radial-basis function interpolation [40] or by recently developed neural networks [44].

The newly developed optimization methods allow for the optimization of highly nonlinear structural problems for applications

that were previously not feasible due to high computational times. The authors plan to use them in the design of heterogeneous materials with high energy dissipation.

### Data availability

The main results of the optimizations in section 3 are provided as csv files in the supplementary data. The results from the brute force computation will be made available on request.

### Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Appendix A. Algorithms

### A.1. Brute-Force search

BF evaluates all designs in the design space  $\mathbb{D}\left(\begin{smallmatrix} N \\ -el \end{smallmatrix}\right)$  for a given grid size  $N_{-el}$  and thus always finds the globally optimal design. Since all designs can be simulated independently, BF can be parallelized very well.

1.  $\text{BF}\left(\begin{smallmatrix} N \\ -el \end{smallmatrix}\right)$
2.  $u_{\min} := \infty$
3. for all  $\mathbf{d} \in \mathbb{D}\left(\begin{smallmatrix} N \\ -el \end{smallmatrix}\right)$  do:
4.  $u := \text{objective}(\mathbf{d})$
5. if  $u < u_{\min}$ :
6.  $u_{\min} := u; \mathbf{d}_{\min} := \mathbf{d}$
7. return  $(u_{\min}, \mathbf{d}_{\min})$

### A.2. Random search

RS randomly draws unique designs  $\mathbf{d} \in \mathbb{D}\left(\begin{smallmatrix} N \\ -el \end{smallmatrix}\right)$  until it finds a design with an endpoint displacement  $u \leq u_{\text{lim}}$ . Then RS immediately returns the current  $u$  and  $\mathbf{d}$ . The computed designs are stored in  $\mathbb{D}_{\text{drawn}}$ . Consequently,  $\mathbb{D}\left(\begin{smallmatrix} N \\ -el \end{smallmatrix}\right) \setminus \mathbb{D}_{\text{drawn}}$  contains only designs that have not been computed yet. Line 4 selects a random design out of these fresh designs.

1.  $\text{RS}\left(\begin{smallmatrix} N \\ -el \end{smallmatrix}, u_{\text{lim}}\right)$
2.  $\mathbb{D}_{\text{drawn}} = \emptyset$
3. for  $n := 1, \dots, \left|\mathbb{D}\left(\begin{smallmatrix} N \\ -el \end{smallmatrix}\right)\right|$  do:
4.  $\mathbf{d} := \text{random}\left(\mathbb{D}\left(\begin{smallmatrix} N \\ -el \end{smallmatrix}\right) \setminus \mathbb{D}_{\text{drawn}}\right)$
5.  $u := \text{objective}(\mathbf{d})$
6.  $\mathbb{D}_{\text{drawn}} := \mathbb{D}_{\text{drawn}} \cup \{\mathbf{d}\}$
7. if  $u < u_{\text{lim}}$ :
8. return  $(u, \mathbf{d})$

### A.3. Genetic algorithm

GA takes as input a list of initial designs  $\mathbf{d}$ , a number of recombinations  $N_{\text{rec}}$ , a mutation probability  $P_{\text{mut}}$ , and a maximum number of iterations without improvement  $N_{\text{cov}}$ . GA uses selection steps,

recombination steps, and mutation steps as described in section 2.2.3 to optimize the design. Note that the cachedObjective function uses a cache that stores previously simulated designs. This prevents surviving designs from being simulated again. When GA computes the objective function in parallel, the parallel workers must wait for each other every iteration before the selection step can be performed. Consequently, GA does not parallelize as well as BF.

1.  $\text{GA}\left(\mathbf{d}, N_{\text{rec}}, P_{\text{mut}}, N_{\text{cov}}\right)$
2.  $N_{\text{pop}} := \text{length}(\mathbf{d})$
3.  $u_{\min} := \infty; n_{\text{cov}} := 0$
4. repeat while not stopped:
5.  $\underline{u} := (\text{cachedObjective}(\mathbf{d}_i))_{1 \leq i \leq N_{\text{pop}}}$
6. if  $u_{\min} > \min(\underline{u})$ :
7.  $u_{\min} := u_j := \min(\underline{u}); \mathbf{d}_{\min} := \mathbf{d}_j$
8.  $n_{\text{cov}} = 0$
9. else if  $n_{\text{cov}} < N_{\text{cov}}$ :
10.  $n_{\text{cov}} := n_{\text{cov}} + 1$
11. else:
12. stop repeat loop
13.  $\underline{f} := (\text{fitness}(u_i))_{1 \leq i \leq N_{\text{pop}}}$
14.  $\underline{\mathbf{d}} := \text{selection}(\underline{\mathbf{d}}, \underline{f})$
15.  $\underline{\mathbf{d}} := \text{recombination}(\underline{\mathbf{d}}, N_{\text{rec}})$
16.  $\underline{\mathbf{d}} := \text{mutation}(\underline{\mathbf{d}}, P_{\text{mut}})$
17. return  $(u_{\min}, \mathbf{d}_{\min})$

### A.4. Simulated Annealing

SA starts with an initial design  $\mathbf{d}$ , a temperature  $T$ , and a cooling rate  $\lambda_T$  that decreases the temperature in  $N_{\text{it}}$  iterations. Note that the name “temperature” is misleading, so  $T$  has neither a kelvin nor a degree Celsius dimension.  $T$  must have the same dimension as the endpoint displacement  $u$  for line 7 to be valid.

The modify( $\mathbf{d}$ )-function randomly flips an element of a design  $\mathbf{d}$ . The flipped element must not lead to a freely movable element. The  $\text{random}(\{\xi \in \mathbb{R} | 0 \leq \xi \leq 1\})$ -function draws a real random number between zero and one.

1.  $\text{SA}(\mathbf{d}, T, \lambda_T, N_{\text{it}})$
2.  $u_{\min} := u' := \text{objective}(\mathbf{d}); \mathbf{d}_{\min} := \mathbf{d} := \mathbf{d}$
3. for  $n := 1, \dots, N_{\text{it}}$  do:
4.  $\mathbf{d} := \text{modify}(\mathbf{d})$
5.  $u := \text{objective}(\mathbf{d})$
6.  $\Delta u := u - u'$
7.  $P_{\text{accept}} := \min(1, \exp(-\frac{\Delta u}{T}))$
8. if  $P_{\text{accept}} \geq \text{random}(\{\xi \in \mathbb{R} | 0 \leq \xi \leq 1\})$ :
9.  $u' := u; \mathbf{d} := \mathbf{d}$
10. if  $u < u_{\min}$ :
11.  $u_{\min} := u; \mathbf{d}_{\min} := \mathbf{d}$
12.  $T := \lambda_T \cdot T$
13. return  $(u_{\min}, \mathbf{d}_{\min})$

### A.5. Free Shape optimization

FS takes an initial design  $\mathbf{d}$  and tries to find better designs by modifying only elements on the edge.

1.  $\text{FS}(\mathbf{d})$
2.  $u_{\min} := \text{objective}(\mathbf{d}); \mathbf{d}_{\min} := \mathbf{d}$

3. for all  $i, j$  do:
4. if  $d_{ij}$  is an edge in  $\mathbf{d}$
5.  $\mathbf{d} := \mathbf{d}$
6.  $d'_{ij} := 1 - d_{ij}$
7.  $u' := \text{objective}(\mathbf{d})$
8. if  $u' < u_{\min}$ :
9.  $u_{\min} := u'; \mathbf{d}_{\min} := \mathbf{d}'$
10. restart line 3
11. return  $(u_{\min}, \mathbf{d}_{\min})$

#### A.6. Breadth-First search with Limited search depth

BFSL examines designs that differ from the current design  $\mathbf{d}$  by at most  $\Delta d_{\max}$  elements. Whenever an improvement is found, BFSL accepts the improved design and restarts evaluating designs that differ from the improved design by at most  $\Delta d = 1$  elements. Line 4 lists all designs that differ by at most  $\Delta d$  elements, and line 5 shuffles this list. Note that the  $\text{vec}(\mathbf{d})$  function makes a vector out of a two-dimensional array  $\mathbf{d}$ , so the 1-norm  $\|\cdot\|_1$  is a vector norm, not a matrix norm.

1. BFSL( $\mathbf{d}, \Delta d_{\max}$ )
2.  $u_{\min} := \text{objective}(\mathbf{d}); \mathbf{d}_{\min} := \mathbf{d}$
3. for  $\Delta d := 1, \dots, \Delta d_{\max}$  do:
4.  $\mathbf{d} := \left( \mathbf{d}' \in \mathbb{D} \mid \Delta d = \|\text{vec}(\mathbf{d}' - \mathbf{d}_{\min})\|_1 \right)$
5.  $\mathbf{d} := \text{shuffle}(\mathbf{d})$
6. for  $i := 1, \dots, \text{length}(\mathbf{d})$  do:
7.  $u_i := \text{objective}(\mathbf{d}_i)$
8. if  $u_i < u_{\min}$ :
9.  $u_{\min} := u_i; \mathbf{d}_{\min} := \mathbf{d}_i$
10. restart line 3 with  $\Delta d := 1$
11. return  $(u_{\min}, \mathbf{d}_{\min})$

#### A.7. Truncated Breadth-First search with Limited search depth

Like BFSL, TBFSL examines designs that differ from the current design  $\mathbf{d}$  by at most  $\Delta d_{\max}$  elements. Furthermore, different elements are separated by at most  $\Delta r_{\max}$ . The distance between different elements  $\Delta r(\mathbf{d}, \mathbf{d}_{\min})$  is computed using the vector 1-norm.

1. TBFSL( $\mathbf{d}, \Delta d_{\max}, \Delta r_{\max}$ )
2.  $u_{\min} := \text{objective}(\mathbf{d}); \mathbf{d}_{\min} := \mathbf{d}$
3. for  $\Delta d := 1, \dots, \Delta d_{\max}$  do:
4.  $\mathbf{d} := \left( \mathbf{d}' \in \mathbb{D} \mid \Delta d = \|\text{vec}(\mathbf{d}' - \mathbf{d}_{\min})\|_1 \wedge \Delta r_{\max} > \Delta r(\mathbf{d}, \mathbf{d}_{\min}) \right)$
5.  $\mathbf{d} := \text{shuffle}(\mathbf{d})$
6. for  $i := 1, \dots, \text{length}(\mathbf{d})$  do:
7.  $u_i := \text{objective}(\mathbf{d}_i)$
8. if  $u_i < u_{\min}$ :
9.  $u_{\min} := u_i; \mathbf{d}_{\min} := \mathbf{d}_i$
10. restart line 3 with  $\Delta d := 1$
11. return  $(u_{\min}, \mathbf{d}_{\min})$

#### A.8. Multi-grid optimization

MG maps the current design  $\mathbf{d}$  to a given problem with size  $N$  using the scale( $\mathbf{d}, N$ ) function and then applies an optimization algorithm  $\mathcal{A}$  to the scaled design.

1. MG( $\mathbf{d}, \left( N_{-1}, \mathcal{A}_1 \right), \left( N_{-2}, \mathcal{A}_2 \right), \dots$ )
2.  $\mathbf{d}_{\min} := \mathbf{d}$
3. for  $\left( N, \mathcal{A} \right) := \left( N_{-1}, \mathcal{A}_1 \right), \left( N_{-2}, \mathcal{A}_2 \right), \dots$  do:
4.  $\mathbf{d}_{\min} := \text{scale}(\mathbf{d}_{\min}, N)$
5.  $(u_{\min}, \mathbf{d}_{\min}) := \mathcal{A}(\mathbf{d}_{\min})$
6. return  $(u_{\min}, \mathbf{d}_{\min})$

#### Appendix B. Supplementary material

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.matdes.2023.111958>.

#### References

- [1] Hans A. Eschenauer, Niels Olhoff, Topology optimization of continuum structures: a review, in: Applied Mechanics Reviews 54.4, July 2001, pp. 331–390. issn: 0003-6900, 2379-0407. doi: 10.1115/1.1388075.
- [2] David J. Munk, Gareth A. Vio, Grant P. Steven, Topology and shape optimization methods using evolutionary algorithms: a review, Struct. Multidisc. Optim. 52 (3) (2015) 613–631.
- [3] Ole Sigmund, Kurt Maute, Topology optimization approaches: a comparative review, Struct. Multidisc. Optim. 48 (6) (2013) 1031–1055.
- [4] Jihong ZHU, Han ZHOU, Chuang WANG, Lu ZHOU, Shangqin YUAN, Weihong ZHANG, A review of topology optimization for additive manufacturing: status and challenges, Chin. J. Aeronaut. 34 (1) (2021) 91–110.
- [5] Mathias Stolpe, Martin P. Bendsøe, Global optima for the Zhou–Rozvany problem, Struct. Multidisc. Optim. 43 (2) (2011) 151–164.
- [6] R. Balamurugan, C.V. Ramakrishnan, Nidur Singh, Performance evaluation of a two stage adaptive genetic algorithm (TSAGA) in structural topology optimization, Appl. Soft Comput. 8 (4) (2008) 1607–1624.
- [7] R. Balamurugan, C.V. Ramakrishnan, N. Swaminathan, A two phase approach based on skeleton convergence and geometric variables for topology optimization using genetic algorithm, Struct. Multidisc. Optim. 43 (3) (2011) 381–404.
- [8] Mark J. Jakiela, Colin Chapman, James Duda, Adenike Adewuya, Kazuhiro Saitou, Continuum structural topology design with genetic algorithms, Comput. Methods Appl. Mech. Eng. 186 (2–4) (2000) 339–356.
- [9] Couro Kane, Marc Schoenauer, Topological optimum design using genetic algorithms, Control. Cybern. 25 (5) (1996) 1059–1088.
- [10] Lester Ingber, Adaptive simulated annealing (ASA): Lessons learned, in: arXiv: cs.MS/0001018 (Jan. 2000). arXiv: cs.MS/0001018.
- [11] Patrick Y. Shim, Souran Manoochehri, Generating optimal configurations in structural design using simulated annealing, in: International Journal for Numerical Methods in Engineering 40.6 (Mar. 1997), pp. 1053–1069. issn: 0029-5981, 1097-0207. doi: 10.1002/(SICI)1097-0207(19970330)40:6<1053::AID-NME97>3.0.CO;2-I.
- [12] Jaesoon Jung, Seongyeol Goo, Junghwan Kook, Design of a local resonator using topology optimization to tailor bandgaps in plate structures, Mater. Des. 191 (2020) 108627.
- [13] Sen Lin, Nengzhuo Chou, Yujia Zhao, Yangfan Qin, Hao Jiang, Junjia Cui, Guangyao Li, Yi Min Xie, Topological optimization of magnetic pulse welding coils with a connectivity-constrained particle swarm optimization algorithm, Mater. Des. 224 (2022) 111337.
- [14] Zhaoyou Sun, Yaguang Wang, Pai Liu, Yangjun Luo, Topological dimensionality reduction-based machine learning for efficient gradient-free 3D topology optimization, Mater. Des. 220 (2022) 110885.
- [15] Yeh-Liang Hsu, A review of structural shape optimization, Comput. Ind. (1994) 11.
- [16] Julian Bauer, Konstantin Priesnitz, Malte Schemmann, Barthel Brylka, Thomas Böhlke, Parametric shape optimization of biaxial tensile specimen: parametric shape optimization of biaxial tensile specimen, Proc. Appl. Math. Mech. 16 (1) (2016) 159–160.
- [17] R. Meske, J. Sauter, E. Schnack, Nonparametric gradient-less shape optimization for real-world applications, Struct. Multidisc. Optim. 30 (3) (2005) 201–218.

- [18] Stefan Riehl, Jan Friederich, Michael Scherer, Ralf Meske, Paul Steinmann, On the discrete variant of the traction method in parameter-free shape optimization, *Comput. Methods Appl. Mech. Eng.* 278 (2014) 119–144.
- [19] Pierre Hansen, Nenad Mladenović, José A. Moreno Pérez, Variable neighbourhood search: methods and applications, in: *Annals of Operations Research* 175.1 (Mar. 2010), pp. 367–407. issn: 0254-5330, 1572-9338. doi: 10.1007/s10479-009-0657-6.
- [20] M.R. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems, *J. Res. Natl. Bur. Stan.* 49 (6) (1952) 409.
- [21] Richard Barrett, Michael Berry, Tony F. Chan, James Demmel, June Donato, Jack Dongarra, Victor Eijkhout, Roldan Pozo, Charles Romine, Henk van der Vorst (Eds.), *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, Society for Industrial and Applied Mathematics, 1994.
- [22] Cihangir Tezcan, Key lengths revisited: GPU-based brute force cryptanalysis of DES, 3DES, and PRESENT, *J. Syst. Archit.* 124 (2022) 102402.
- [23] Christian Blum, Andrea Roli, Metaheuristics in combinatorial optimization: overview and conceptual comparison, *ACM Comput. Surv.* 35 (3) (2003) 268–308.
- [24] Amir Hossein Gandomi, Xin-She Yang, Siamak Talatahari, Amir Hossein Alavi, in: *Metaheuristic Applications in Structures and Infrastructures*, Elsevier, 2013, pp. 1–24.
- [25] Amin Riazi, Genetic algorithm and a double-chromosome implementation to the traveling salesman problem, *SN Appl. Sci.* 1 (11) (2019).
- [26] Carlos A Coello Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, *Comput. Methods Appl. Mech. Eng.* 191 (11-12) (2002) 1245–1287.
- [27] Tim Head et al., *scikit-optimize/scikit-optimize*, Oct. 2021. doi: 10.5281/ZENODO.5565057.
- [28] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, Nando de Freitas, Taking the human out of the loop: a review of Bayesian optimization, *Proc. IEEE* 104 (1) (2016) 148–175.
- [29] B Suman, P Kumar, A survey of simulated annealing as a tool for single and multiobjective optimization, *J. Oper. Res. Soc.* 57 (10) (2006) 1143–1160.
- [30] Shu-Lin Wang, Xue-Ling Li, Jianwen Fang, Finding minimum gene subsets with heuristic breadth-first search algorithm for robust tumor classification, *BMC Bioinf.* 13 (1) (2012).
- [31] Jinwoo Seok, Pierre Kabamba, Anouck Girard, Task selection for radar resource management in dynamic environments, *J. Eng.* 2018 (1) (2018) 1–9.
- [32] N. Mladenović, P. Hansen, Variable neighborhood search, *Comput. Oper. Res.* 24 (11) (1997) 1097–1100.
- [33] Nenad Mladenovic et al., Variable Neighborhood Search: 8th International Conference, ICVNS 2021, Abu Dhabi, United Arab Emirates, March 21–25, 2021, *Proceedings*, vol. 12559, Springer Nature, 2021.
- [34] Damir Sedlar, Zeljan Lozina, Ivan Tomac, Discrete optimization of truss structures using variable neighborhood search, in: *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, 2021. Publisher: Springer, pp. 1–16.
- [35] Wolfgang Hackbusch, *Multi-Grid Methods and Applications*. en. Vol. 4. Springer Series in Computational Mathematics, Springer Berlin Heidelberg, Berlin, Heidelberg, 1985. isbn: 978-3-662-02427-0. doi: 10.1007/978-3-662-02427-0.
- [36] Victor A. Paludetto Magri, Robert D. Falgout, Ulrike M. Yang, A New Semi-Structured Algebraic Multigrid Method. en. arXiv:2205.14273 [cs, math]. May 2022.
- [37] M. Brown, R. Szeliski, S. Winder, Multi-image matching using multi-scale oriented patches, in: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, IEEE, San Diego, CA, USA, 2005, pp. 510–517. isbn: 978-0-7695-2372-9. doi: 10.1109/CVPR.2005.235.
- [38] Jinghua Zhang, Chen Li, Frank Kulwa, Xin Zhao, Changhao Sun, Zihan Li, Tao Jiang, Hong Li, Shouliang Qi, A multiscale CNN-CRF framework for environmental microorganism image segmentation, *Biomed Res. Int.* 2020 (2020) 1–27.
- [39] Vasilis Krokos et al., A Bayesian multiscale CNN framework to predict local stress fields in structures with microscale features. en, in: *Computational Mechanics* 69.3 (Mar. 2022), pp. 733–766. issn: 0178-7675, 1432-0924. doi: 10.1007/s00466-021-02112-3.
- [40] Sujin Bureerat, Jumlong Limtragool, "Structural topology optimisation using simulated annealing with multiresolution design variables". en, *Finite Elem. Anal. Des.* 44 (12-13) (2008) 738–747.
- [41] Norman Lloyd Johnson, Adrienne W. Kemp, Samuel Kotz, *Univariate Discrete Distributions*, third ed., Wiley, Hoboken, NJ., 2005. isbn: 978-0-471-27246-5.
- [42] Pauli Virtanen et al., SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, in: *Nature Methods* 17, 2020, pp. 261–272. doi: 10.1038/s41592-019-0686-2.
- [43] David W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*. en. 1st ed. Wiley Series in Probability and Statistics. Wiley, Aug. 1992. isbn: 978-0-471-54770-9. doi: 10.1002/9780470316849.
- [44] Saketh Sridhara, Aaditya Chandrasekhar, Krishnan Suresh, A generalized framework for microstructural optimization using neural networks, *Mater. Des.* 223 (2022) 111213.

---

**Paper D**

# Optimizing the Hole Geometry of 2d Plates for Maximum Tensile Toughness

Authors: Rettl, Matthias  
Pletz, Martin  
Schuecker, Clara

Preprint. Under Review in Engineering Fracture Mechanics

DOI: [10.2139/ssrn.4962118](https://doi.org/10.2139/ssrn.4962118)



# Optimizing the hole geometry of 2D plates for maximum tensile toughness

Matthias Rettl, Martin Pletz\*, Clara Schuecker

*<sup>a</sup>Designing Plastics and Composit Materials, Department of Polymer Engineering and Science, Montanuniversität Leoben, Otto-Gloeckel-Straße 2, Leoben, 8700, Steiermark, Austria*

---

## Abstract

A tough material response is important in many fields and can be achieved for brittle materials by adding holes. In this work, a FEM approach is presented to maximize the tensile toughness of pre-cracked 2D plates by adding arbitrarily shaped holes. The initial crack is stopped by a hole and a new crack must initiate at a higher load. This fracture process is predicted using Taylor's Point Method and Griffith's criterion, which is estimated by Configurational Forces. The toughest plate, optimized in a level-set like approach, achieves a tensile toughness 4.5 times higher than a solid plate in all load directions.

*Keywords:* Fracture mechanics, finite element method, crack paths, optimization

---

## 1. Introduction

Common engineering materials are usually either strong or tough [1]. Most applications, however, require a material that is both strong and tough but still retain an acceptable stiffness; where strength is either the yield stress or the ultimate stress, and toughness is either the resistance to crack propagation or the total energy to failure (the integral of the  $\sigma(\varepsilon)$  curve) [2].

Some natural materials are both strong and tough due to their heterogeneous, hierarchical structure [3, 4]. The toughening effect is a result of the combination of strong but brittle base materials with ductile materials of lower stiffness and yield stress. Depending on the mechanical properties of the two materials and the interface between them as well as the geometrical arrangement of the phases on the various hierarchical levels, additional damage mechanisms can be activated such as crack deflection, crack trapping,

---

\*martin.pletz@unileoben.ac.at

<b>abbreviations</b>	<b>symbols</b>
PM Point Method	$\mathbf{x}, \mathbf{x}_{\text{mid}}$ position vector, center of the plate
CC Coupled Criterion	$\mathbf{n}$ normal vector
LM Line Method	$n$ a number
PMMA Poly(methyl methacrylate)	$\sigma, \sigma_{\text{I}}$ stress, max. principal stress
ABS Acrylonitrile butadiene styrene	$\varepsilon$ strain
FEM Finite element method	$F$ force
RBF Radial basis function	$u, u_{\text{max}}, u_{\text{load}}$ displacement, max. applied displacement, applied displacement
TBFSL Truncated breadth first search with limited depth	$\tilde{u}$ dimensionless load factor
VNS Variable neighborhood search	$S$ stiffness
<b>nomenclature</b>	$E$ Young's modulus
$q$ scalar quantity	$\nu$ Poisson's ratio
$\mathbf{q}$ physical vector (2. order tensor)	$K_{\text{I}}, K_{\text{Ic}}$ stress intensity factor, fracture toughness
$\vec{q}$ sequence	$G$ energy release rate
<b>sub- and superscripts</b>	$\mathbf{J}$ vectorial J-Integral
$q$ corresponds to:	$l_{\text{ch}}$ material's characteristic length
$q_{\text{FEM}}$ a FEM simulation	$U_{\text{T}}$ tensile toughness
$q_{\text{ini}}$ crack initiation	$U_{\text{T}}^{\text{min}}$ minimum tensile toughness for the rotated plates
$q_{\text{c}}$ a critical value for crack propagation or initiation	$a, a_0$ crack length, initial crack length
$q_{\text{s}}$ support points	$\Delta\alpha$ crack kink angle in a crack increment
$q_{\text{f}}$ fixed support points	$l$ side length of plate
$q_{\text{v}}$ variable support points	$\varphi$ rotation angle of plate
$q_{\text{grid}}$ the grid for the variable support points	$d, l_{\text{d}}, l_{\text{low}}, l_{\text{high}}$ dimensions of the support points
$q_{\text{worker}}$ asynchronous worker processes	$f$ RBF function value
$q^*$ the optimal design	$f_{\text{t}}$ threshold value
$q^{(i)}$ the $i$ -th optimization iteration	$w_i$ weight of $i$ -th support point for the RBF interpolation
$q^{[k]}$ the $k$ -th crack increment in a fracture simulation	$l_{\text{rbf}}$ length scale for RBF kernel
$\Delta q$ a crack increment	$\mathbb{X}$ design space
$q_{\text{c},i}^{\text{G}}$ the critical load for crack propagation of the $i$ -th crack according to LEFM	$\Gamma$ all hole contours
$q_{\text{c},j}^{\sigma}$ the critical load for initiation at $j$ -th hole according to PM	$\Gamma_{h,i}$ hole contour of $i$ -th hole
	$d_{\text{TBFSL}}$ parameter of TBFSL optimization algorithm
	$r_{\text{TBFSL}}$ parameter of TBFSL optimization algorithm

secondary crack initiation, friction between microcrack surfaces, and crack bridging. In addition, the hierarchical structure can limit the reduction in failure stress caused by initial flaws, making the structure more damage tolerant.

These mechanisms have been investigated in natural materials such as nacre [5, 6], deep sea sponges [7], and bone [8]. Some of these mechanisms are already employed in engineering structures such as layered composites [9, 10], materials with architected micro-structure produced by high plastic deformation [11], block-copolymers such as ABS [12], or additively-manufactured structures [13]. In order to develop mi-  
crostructures for even tougher materials, a fundamental understanding of these damage mechanisms – and  
parameters controlling them – is crucial. In natural or artificial heterogeneous materials, however, many of  
those mechanisms can occur at the same time and possibly on several length scales.

An example of simplified versions of heterogeneous materials are layered structures, which have been  
shown to efficiently stop cracks due to the effects of inhomogeneous stiffness [10], inhomogeneous yield  
stress [14, 15], or residual stresses [16]. Once a crack has been arrested, it needs a higher applied load to  
propagate (or re-initiate). An extreme case of layers with inhomogeneous stiffness has been addressed by  
[17] considering a parallel arrangement of separate, unconnected sheets where the small gap between the  
sheets can be considered as a material with zero stiffness. A layered structure, however, only works for  
stopping transverse cracking of layers, while cracks parallel to the layers can grow fairly unhindered.

For general applications with undefined crack orientations, a more suitable topology is the use a softer  
second phase in the form of particles inside a matrix, as known e.g. for ABS [12] or toughened epoxies  
[18, 19]. These particles work for multiple crack directions: the particles can first attract the crack, and  
then require an increased load for a crack initiating from them. Furthermore, several cracks may initiate,  
but as long as they have not propagated through the second phase, the material is not completely separated.  
In addition to the material properties of the particles and interface properties, the toughening effect in such  
materials is also driven by the particle geometry.

For a thorough understanding of the contribution of the particle geometry, it needs to be studied indepen-  
dently, which can be achieved by replacing the particles by holes to discount the effect of the interface and  
property mismatch. Such voids can divert or trap cracks, such that new cracks need to initiate at these holes  
which requires higher applied stresses. This has been studied in 2D setups (using holes instead of voids)



---

experimentally with additively manufactured specimens [20]. Numerical studies of such setups [21, 22] focus on predicting in which cases cracks are trapped, but does so without considering re-initiation of cracks.

This work studies the effect of the hole geometry in a Finite Element Method (FEM) model that considers both crack propagation and crack initiation. To this end, an optimization strategy is developed and implemented to determine hole patterns that maximize the tensile toughness of a 2D plate containing an initial crack. The plate can contain multiple arbitrary-shaped holes which are adapted by the optimization algorithm. The aim is to find hole patterns that maximize the toughness for several possible initial cracks (in this case in four different directions are considered) to avoid layered-type patterns which would be optimal if only one crack direction was considered.

## 2. Methods

To study the toughening effect of hole geometry, a 2D plate with an arbitrary arrangement of holes is considered (see Fig. 1a). The plate contains an initial crack loaded in tension (Mode I) and is simulated using the FEM Package ABAQUS Standard [23]. All FEM analyses are purely linear (linear elastic material, linear geometry, small displacement). Crack propagation and initiation of new cracks (without dynamic effects), however, is captured by automatically generating and analyzing FEM models with updated crack increments following the *full FEM* scheme of a previous work [24] in combination with the Point Method [PM] [25] as crack initiation criterion. Therefore, the model can contain any number of cracks that can propagate in the bulk material, initiate from a hole, or get arrested by growing into a hole surface, the specimen surface, or another crack. In each increment, only one crack can propagate or initiate at a time. Each hole pattern is analyzed in 4 different configurations to account for cracks approaching from 4 different sides. The tension load for each configuration is applied by a prescribed incremental displacement  $u$ . From the incremental FEM computations (including incremental crack propagation until failure of the whole plate), the reaction force  $F(u)$  is obtained, from which the tensile toughness of the plate,  $U_T$  is then computed as the integral of  $F(u)$  over the displacement (as also done in [2]).

With such a model, the hole geometry can be optimized for maximum  $U_T$ . With only one setup with a

crack on the left of the plate, this would result in design similar to a layered plate or other trivial solutions (such as a arc-shaped hole in front of the crack tip [26]). Therefore, the initial crack is sequentially placed on each side of the square. To optimize the geometry of the holes, support points are aligned in a grid. Each support point has a parameter, which can be either 1 (material) or 0 (hole). The parameters are interpolated  
 75 and the contour line of this interpolation defines smooth hole shapes that are added to the plate. Additionally, the design space is simplified by considering a) x- and y-symmetry and b) rotational symmetry. A previously developed optimization algorithm [27] is used to optimize the design.

### 2.1. FEM model setup

The setup of the FEM model is illustrated in Fig. 1a. The square plate has a side length  $l = 80$  mm  
 80 and contains an initial crack located at half of its height with a length  $a_0 = 10$  mm. The holes can have any shapes defined by splines, but must not intersect the initial crack or the outer edges of the plate. The material is assumed as linear elastic with parameters given in Sapora et al. [28] for PMMA with  $E = 2960$  MPa,  $\nu = 0.38$ .  $K_{Ic} = 1.96$  MPa  $\sqrt{\text{m}}$ , and  $\sigma_c = 70.5$  MPa. The model uses linear geometry and plane strain elements CPE4R and CPE3 with an assigned thickness of 1 mm (to directly obtain output in consistent units).

85 The model uses two types of cracks: *active* cracks have a crack tip and can propagate, whereas *inactive* cracks already ended in a hole surface, the plate surface, or another crack. The areas around active cracks are partitioned using a tube of radius 0.75 mm, as done in [24]. The mesh is refined towards the crack tips from a global mesh size of 2 mm to a local mesh size of 0.15 mm, see Fig. 1b. The factor of the local mesh  
 90 size to the tube radius is chosen according to a mesh size study in [29], where this mesh size has been shown to be sufficiently fine. Hole surfaces are meshed with an element edge length of 1 mm if the model contains active cracks. If the model does not contain active cracks, hole surfaces are meshed with an edge length of 0.45 mm to accurately predict crack initiation from a hole surface, which mainly occurs if no active cracks are present. The bottom edge of the plate is constrained in the vertical direction. The nodes of the upper  
 95 edge are coupled in y-direction to a reference point, at which a displacement  $u_{\text{FEM}} = 1$  mm is applied and the reaction force  $F_{\text{FEM}}$  is output. Note, that the upper edge has a uniform vertical displacement and cannot rotate.

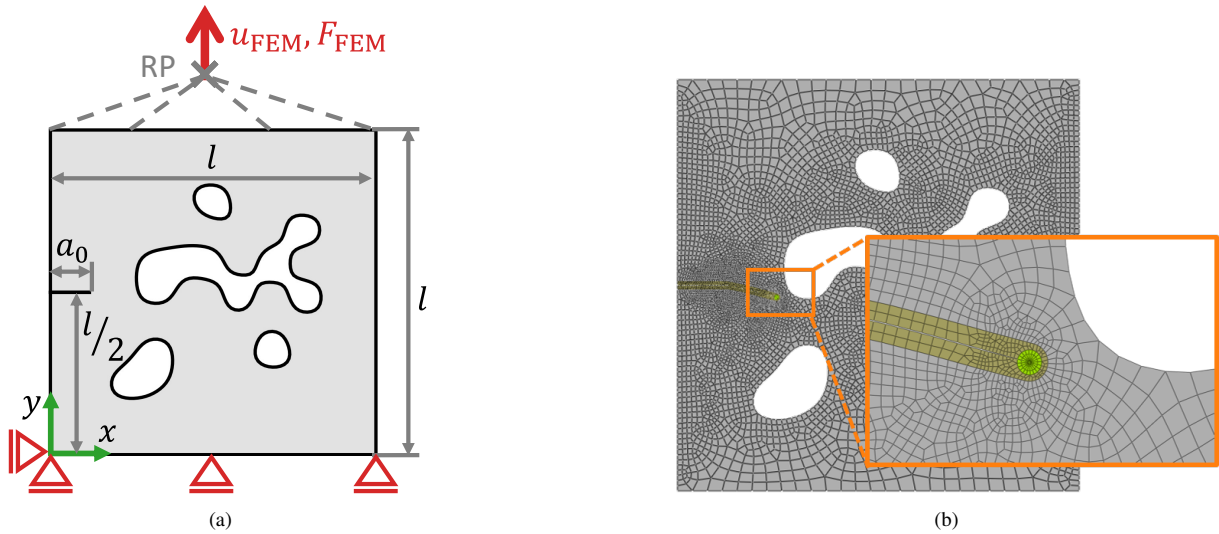


Figure 1: Model sketch for the used 2D model and the corresponding mesh that is refined towards the crack tips.

In the FEM model, cracks incrementally propagate with a fixed increment length  $\Delta a$ , see Section 2.2.1.  
 100 If initiation from a hole surface is predicted (for details, see Section 2.2.2), cracks are inserted with a starting crack length  $\Delta a_{\text{ini}} = 1 \text{ mm}$ . The FEM model is repeatedly computed with an updated and remeshed crack geometry.

The crack propagation increment length  $\Delta a$  necessary for obtaining accurate results (particularly in  
 105 terms of accurate crack paths) is obtained by varying  $\Delta a$  in six selected hole designs, depicted in Fig. 2. These designs showed to be sensitive to varied  $\Delta a$  in preliminary computations. The smallest  $\Delta a$  needed for accurate predictions in these designs is considered to be sufficient and is used in all further model runs independent of the chosen grid size for the optimization. The results of the parameter study are presented in Section 3.1.

## 110 2.2. Modeling cracks

### 2.2.1. Incremental crack propagation

The propagation of an active crack is determined from the vectorial energy release rate  $\mathbf{J}$ , which is based on configurational forces evaluated inside the tubes surrounding the crack tip. The direction of  $\mathbf{J}$  is taken as the propagation direction and its magnitude as the scalar energy release rate  $G$ . The limitations for

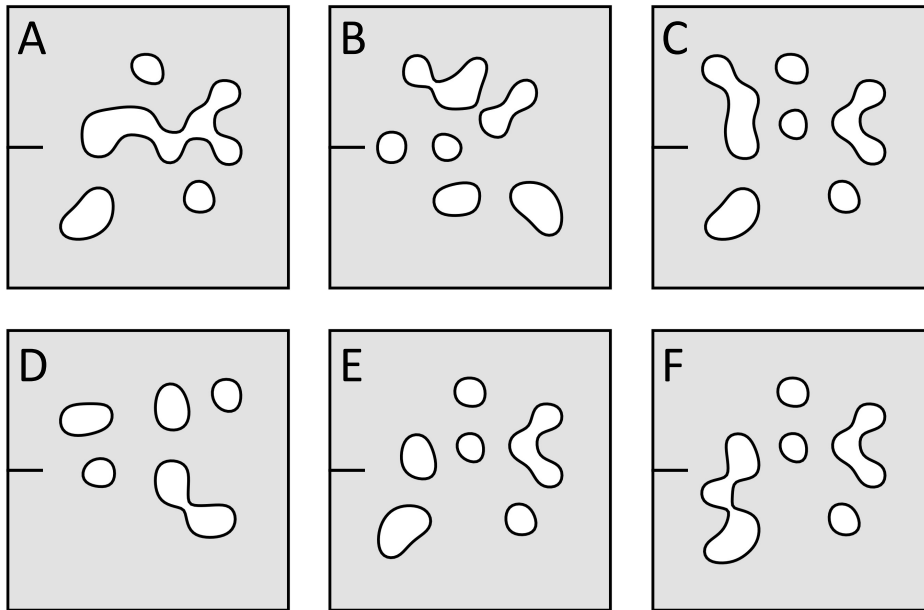


Figure 2: Six hole designs A-F used for identifying the  $\Delta a$  necessary for accurately predicting crack paths.

115 large mode II to mode I ratios of these methods have been discussed in [24], where efficient approaches for incremental crack propagation are suggested. In this work, we use the *explicit* approach described therein with crack propagation defined by a linear increment in the direction of  $\mathbf{J}$ , without any angle corrections. Essentially, the modeling described in [24] is used in this work with the full  $\mathbf{J}$  evaluation code provided and documented in [30]. A small adaptation is introduced for dealing with crack tips close to a hole surface,  
 120 as shown in Fig. 3: instead of increasing  $\Delta a$  in the last increment before the crack runs into the hole or passes it, further increments using the angle change  $\Delta\alpha$  from the last increment are added. This has proven to perform better in cases where the crack tip comes close to a hole surface and might or might not pass it. Active cracks in the model are meshed more finely close to their crack tips. For details on this mesh and partitions, the reader is referred to [24]. Once a crack runs into either a hole surface, an outer boundary of  
 125 the plate, or another crack, the crack is deactivated and its mesh size is reduced to the global mesh size in the model.

For checking if a crack runs into a hole surface, the spline points of the hole surface are discretized as a poly-line. This is more robust than computing the intersection of an extended crack and a spline, but means

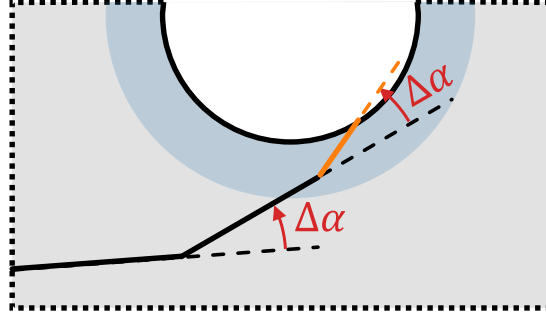


Figure 3: If a potential extended crack tip lies too close to a hole surface (in the blue region), it is extended using additional crack increments with length  $\Delta a$  and an angle change of  $\Delta\alpha$  each. This is repeated until the crack has grown into the hole or its tip does not lie too close to a surface any more.

130 that the points approximating the spline must be rather fine. Furthermore, we check if a crack runs into any other crack.

### 2.2.2. Crack initiation from holes

For holes with a very small curvature radius  $\rho$ , the classical stress criterion no longer applies. In the extreme case of a curvature radius  $\rho = 0$ , the stress approaches infinity and a fracture-mechanical criterion  
 135 must be used. Approaches that capture both large and very small  $\rho$  have been developed, such as the Coupled Criterion (CC) of Leguillon [31], which is based both on stresses and the incremental energy release rate of a possibly initiating crack with length  $a$ . To avoid the additional computations necessary for obtaining the incremental energy release rate, faster methods have been developed using a matched asymptotics approach [31] or multi-scale computations [32]. Taylor [25] proposed two such fast methods based solely on stress,  
 140 called Point Method (PM) and Line Method (LM). In this work, the PM is used. The PM takes the maximum in-plane principal stress  $\sigma_I$  at a normal distance  $l_{ch}/2$  from the surface and compares it to the critical stress  $\sigma_c$  of the material. The critical energy release rate  $G_c$  of the material is accounted for by the characteristic length

$$l_{ch} = K_{Ic}^2 / (\pi \sigma_c^2) \quad (1)$$

with

$$G_c = K_{Ic}^2 (1 - \nu^2) / E \quad (2)$$

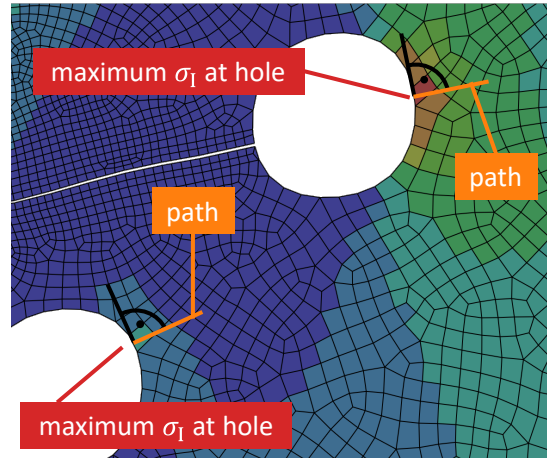


Figure 4: Mesh-based evaluation of the crack initiation position  $\mathbf{x}_{ini}$  and normal for the initiation  $\mathbf{n}_{ini}$  in the element with the highest  $\sigma_1$  at a hole surface.

145 for plane strain. For the PMMA material parameters stated above,  $G_c = 1110 \text{ J/m}^2$  and the characteristic length for plane strain is  $l_{ch} = 0.246 \text{ mm}$ .

In the FEM model, we evaluate  $\sigma_1$  at the element integration points along the surface of each hole and find its maximum for each hole, as illustrated by Fig. 4, where it is assumed that a crack initiates at the position where  $\sigma_1$  is maximum. This initiation position,  $\mathbf{x}_{ini}$ , is evaluated based on the mesh: The mid-side node of the element edge is taken as the initiation position  $\mathbf{x}_{ini}$  and the normal to this edge defines the initiation direction  $\mathbf{n}_{ini}$ .

The PM of Taylor requires the stress along the path of a potentially initiating crack. This stress is extracted from FEM results at a distance of  $l_{ch}/2$  from the surface. With reasonable element size for the optimization, this evaluation point always lies within the first element. Since the path evaluation of Abaqus first averages result variables at nodes before interpolating them, linear elements yield inaccurate path results within the element next to the surface. The maximum stress evaluated in this way yields the same value as the maximum stress evaluated at an integration point, which considerably depends on the mesh size in case of a steep stress gradient. To avoid using very fine mesh sizes, we use a simple method to improve the accuracy of the stresses evaluated close to surfaces as follows.

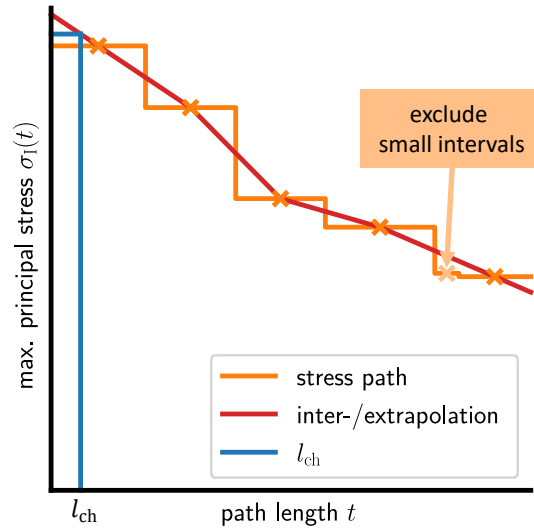


Figure 5: Interpolation and extrapolation of  $\sigma_1$  along the path for obtaining mesh-independent  $\sigma_1$  values at the characteristic length  $l_{ch}$ . The middle of each element cut is used with the corresponding element  $\sigma_1$  value, but omitting elements that are only slightly cut.

This method evaluates  $\sigma_1$  along a path that starts at the defined initiation position at the hole and points into the material normal to the surface. Stresses are read from each element intersected by the path and the  $\sigma_1$  stress along the path length  $t$  is computed at the mid-point of each path section of a specific element as shown in Fig. 5. The stress along the path is then computed by linear interpolation between those mid-points and extrapolation to the hole surface in the first element along the path. Note, that for very short path sections, which occur if an element is just barely intersected by the path, the contribution of such a section is ignored by excluding the corresponding  $\sigma_1$  value from the interpolation. This is done for path sections with a length smaller than 0.4 of the longest interval as indicated in Fig. 5 for the second-last path section. The resulting stress profile is then used to evaluate  $\sigma_1(l_{ch}/2)$  to assess crack initiation in terms of the necessary load factor  $\tilde{u}_c = u_c/u_{FEM}$  for each hole with its critical applied displacement  $u_c$ . It is noted, that more accurate solutions for path interpolation near hole surfaces could be devised, however, for our purposes (where initiation is usually triggered by the stress data from one of the first two elements), this method has been found to be robust and more realistic than the method offered by Abaqus.

### 2.2.3. Combining crack propagation and crack initiation

In the linear FEM model with applied  $u_{\text{FEM}}$ , the stresses and the energy release rates  $G$  can be scaled by  $\tilde{u}$  as  $\sigma = \tilde{u} \sigma_{\text{FEM}}$  and  $G = \tilde{u}^2 G_{\text{FEM}}$  for any other load  $u = \tilde{u} u_{\text{FEM}}$ . This also means that for all crack tips with index  $i$ ,  $G_i$  can be used to obtain corresponding critical loads  $u_{c,i}^G$  ( $i \in 1, \dots, n_{\text{cracks}}$ ) for the propagation criterion  $u_{c,i}^G/G_c = 1$ . For all holes with index  $j$ ,  $\sigma_{I,j}(l_{\text{ch}}/2)$  can be used to obtain the corresponding critical loads  $u_{c,j}^\sigma$  ( $j \in 1, \dots, n_{\text{holes}}$ ) for the initiation criterion  $u_{c,j}^\sigma/\sigma_c = 1$ . The incremental FEM model evaluates all  $u_c$  values in each increment, and takes the lowest  $u_c$  to either propagate or initiate a crack. With the updated model,  $u_c$  values are evaluated and cracks either propagate or initiate. From one increment to the next, the lowest  $u_c$  can either decrease (which is common for a growing crack) or increase (which can happen when a crack grows into a hole and then needs to reinitiate or in cases of stable crack growth). This procedure is repeated until the plate is broken and has zero stiffness or  $u_{\text{load}}$  reaches the maximum displacement defined as  $u_{\text{max}} = 2$  mm. Almost all designs break at lower displacements and have zero force before they reach  $u_{\text{max}}$ . The remaining designs are almost fully broken at  $u_{\text{max}}$ , but they have a small material bridge with a very low stiffness that still connects the plate. Since the contributions to  $U_T$  of these bridges are very small, it can be neglected. To obtain the  $F(u)$  curve of the plate, in each increment  $k$ , the lowest  $u_c^{[k]}$  and the plate stiffness  $S^{[k]}$  (which decreases with increasing  $k$ ) are extracted. The  $F(u)$  curve is built from a sequence of  $(u_{\text{load}}^{[k]}, S^{[k]})$  points, starting with  $u_{\text{load}}^{[0]} = 0$ . For each increment,  $u_{\text{load}}$  is updated if  $u_c^{[k]}$  is larger than the previous  $u_{\text{load}}$ .

An example of this incremental crack propagation and initiation is illustrated in Fig. 6. The plate with one hole at its center, depicted in Fig. 6c, initially has one crack that can propagate and one hole, a crack can initiate from. Figure 6a shows the corresponding  $u_c$  values for crack growth (blue line) and initiation (orange x markers). In the initial crack propagation increment ( $k = 0$ , stage 1 in green in Fig. 6)  $u_{\text{load}}$  increases to 0.25 mm, which is the initial  $u_c$  for crack propagation. In the next increments,  $u_c$  of the crack is below this  $u_{\text{load}} = 0.25$  mm, which means that the crack just grows at this constant  $u_{\text{load}}$  with a drop in force, as can be seen in Fig. 6b and is indicated as stage 2. In increment  $k = 12$ , the crack grows into the hole, so at this point, there are no active cracks in the model. Therefore, the load for initiation of  $u_c = 0.45$  mm is most critical and  $u_{\text{load}}$  is increased to this value (stage 3 in Fig. 6), leading to a force increase of about 800 N.



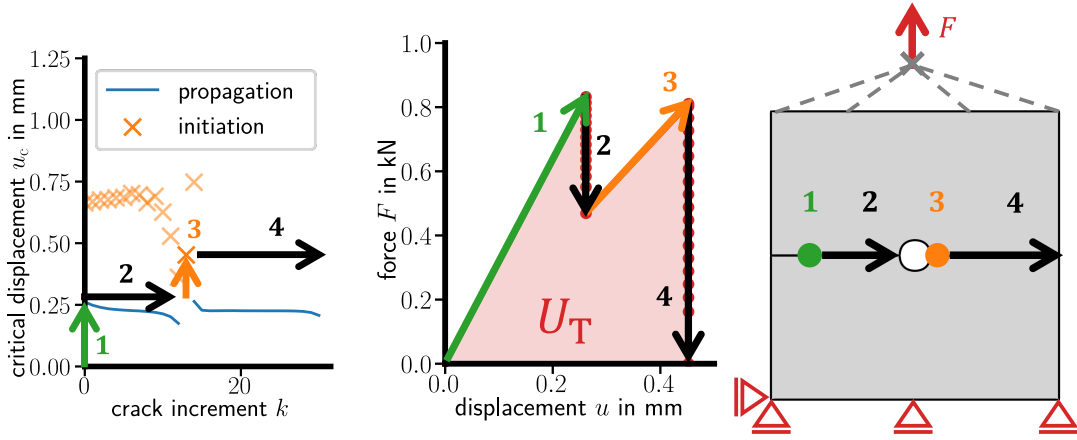


Figure 6: Evaluation of a FEM model with a central hole. In each increment, the critical  $u_c$  for crack propagation and crack initiation are plotted. Using those  $u_c$ , the  $F(u)$  curve can be extracted, which featured kinks where the crack stops and propagates/initiates at an increased  $u_c$ . Four stages are identified: 1) load increase until propagation if initial crack, 2) propagation of the initial crack into the hole, 3) increase of load until a crack initiates from the hole, and 4) initiation and critical propagation of crack from the hole.

205 Once this  $u_{\text{load}}$  is reached, a crack initiates at the right side of the circular hole, and since  $u_c$  is already much higher than needed for this crack to grow further, it propagates unstably through the rest of the plate (stage 4 in Fig. 6). This is associated with a drop of the force to 0. From the  $F(u)$  curve, the tensile toughness  $U_T = \int_0^{u_{\text{max}}} F(u) du$  can be computed (which corresponds to the area under the  $F(u)$  curve up to  $u_{\text{max}}$ ).

### 2.3. Definition of hole designs

210 In order to find an optimal design,  $U_T$  should be maximized by varying the geometry of holes in the plate. To vary the hole geometry, a general representation of hole shapes is required that allows for creating new holes, deleting existing holes, or changing the shape of existing holes. For this, an approach similar to parameterized level-set methods [33, 34, 35] is used, which defines a smooth parameterized function  $f(\mathbf{x}, \dots)$ , whose contour line at a fixed threshold  $f_t$  defines the holes. Figure 7 shows such a smooth function  $f(\mathbf{x}, \dots)$  that interpolates so called support points (spheres and cubes) using Radial Basis Functions (RBF) [36]. The support points are aligned in a grid and have either a low level to represent material or a high level to represent holes. The function  $f(\mathbf{x}, \dots)$  is generated by interpolating the support points using RBF, which are used in many fields of application like for gaussian processes [37] and response surface methods for surrogate models [38], because RBF can interpolate complex functions smoothly. All function values above the  
220 threshold  $f_t$  (which will be set to 0.5) are holes (green) and all function values below  $f_t$  are material (orange).

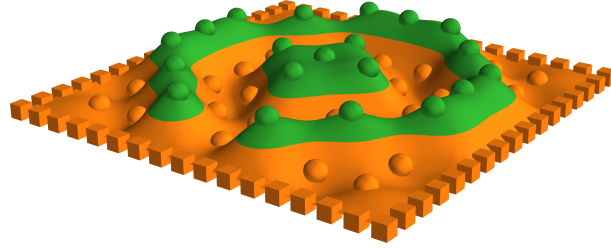


Figure 7: Interpolation of support points (spheres and cubes) using Radial Basis Functions. The interpolation is drawn orange for values  $< f_i$  (material) and green for values  $\geq f_i$  (hole).

For optimization, the algorithm Truncated Breadth-First Search with Limited depth (TBFSL) [27] is used since it does not need derivatives and is able to escape local optima. Note, that the position of the support points is held constant during an optimization run. However, the position of the support points might influence the result, because a hole contour can never cross a support point due to the values of 0 or 1 defined at those points. To compensate this drawback, we perform multiple optimization runs with varied support point spacing  $2d$ , see Section 2.4.4.

### 2.3.1. The design space

Figure 8 depicts the specimen with the support points. Two types of support points are used. Fixed support points  $\vec{\mathbf{x}}_f := (\mathbf{x}_{f,1}, \dots)$  are marked in blue and form a rectangle that defines the perimeter of the design space with internal variable support points  $\vec{\mathbf{x}}_v := (\mathbf{x}_{v,1}, \dots)$  marked in orange. The variable support points are arranged in a  $n_{\text{grid}} \times n_{\text{grid}}$  grid, amounting to  $n_v = n_{\text{grid}} \cdot n_{\text{grid}}$  points. The grid size is defined for each optimization run. The design space

$$\mathbb{X} := \left\{ \mathbf{x} := (x, y) \left| \begin{array}{l} l_{\text{low}} \leq x \leq l_{\text{high}} \\ \wedge \\ l_{\text{low}} \leq y \leq l_{\text{high}} \end{array} \right. \right\} \quad (3)$$

thereby contains every point  $\mathbf{x} = (x, y)$  inside the rectangle formed by the fixed support points  $\vec{\mathbf{x}}_f$ . All support points  $\vec{\mathbf{x}}_s := (\mathbf{x}_{f,1}, \dots, \mathbf{x}_{v,1}, \dots)$  are used to parameterize the shape of holes, with a total number of support points  $n_s$ . For this, binary values  $\vec{f}_s := (f_{s,1}, \dots) \in \{0, 1\}^{n_s}$  are assigned to each of the support points. A value of  $f_s = 0$  stands for a point inside the material and a value of  $f_s = 1$  stands for

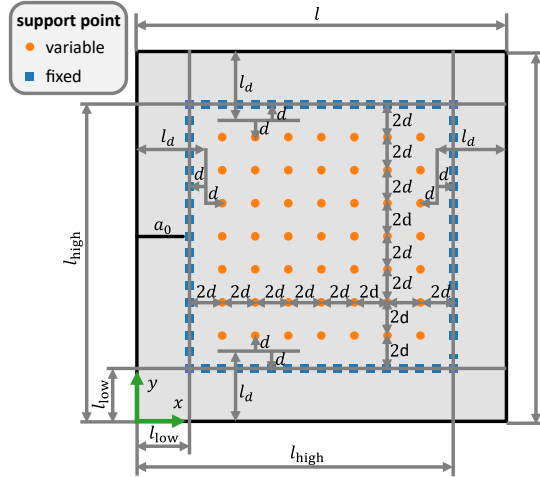


Figure 8: The optimization uses support points  $\vec{x}_s$  that can be variable (orange) or fixed (blue). The design space  $\mathbb{X}$  contains every point  $\mathbf{x}$  inside the rectangle defined by the fixed support points.

a point inside a hole. To ensure that holes can only occur inside the design space  $\mathbb{X}$ , the fixed support points  $\vec{x}_f$  are assigned to zero values  $f_f^{\vec{x}} := (0, \dots)$  (i.e. material). Accordingly, the support point values  
 240  $\vec{f}_s^{\vec{x}}(f_v^{\vec{x}}) := (\vec{f}_f^{\vec{x}}, \vec{f}_v^{\vec{x}}) = (0, \dots, f_{v,1}, \dots)$  depend only on the variable vector  $f_v^{\vec{x}}$  corresponding to the variable support points.

As depicted in Fig. 8, the mid-point between a fixed support point and the first variable support point has a distance of  $l_d = 15$  mm from the specimen edge. Since the boundaries of holes are obtained from  
 245 interpolations of  $f(\mathbf{x}, \dots)$ , this distance is approximately the minimum distance of hole boundaries from the specimen edge. The distance  $d$  is computed as  $2d = \frac{(l-2 \cdot l_d)}{n_{\text{grid}}}$  and depends on the grid size  $n_{\text{grid}}$  of the variable support points. The boundaries of the design space  $\mathbb{X}$  are then computed as:

$$\begin{cases} l_{\text{low}} := l_d - d \\ l_{\text{high}} := l - l_d + d \end{cases} \quad (4)$$

The following paragraphs explain the key elements to define the hole geometry inside the design space. As already mentioned, the function  $f(\mathbf{x}, \dots)$  uses *RBF* to interpolate between the support points. The RBF  
 250 interpolation requires a so called *length scale* that defines the area of influence of the support points. Next, a *threshold*  $f_t$  is applied to  $f(\mathbf{x}, \dots)$  to generate contour lines that represent the hole surface. Furthermore, the contour lines might contain invalid material islands. This is checked by a *validation* function. Finally,

a crack growth should not only be investigated for one direction, but for several directions. For this, the geometry is transformed by a *rotation* function.

255 *RBF*. An example of a RBF interpolation is shown in Figure 7. The support points are interpolated using a Gaussian RBF kernel [37]

$$k(\mathbf{x}, \mathbf{x}') := \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{l_{\text{rbf}}^2}\right), \quad (5)$$

which describes the interdependency of two points  $\mathbf{x}, \mathbf{x}'$ . The length scale  $l_{\text{rbf}}$  is chosen such that the interpolation is smooth, but does not overshoot too much. Overshooting is the phenomenon when an interpolation value reaches values higher or lower than values of its neighboring support points. The function

$$f(\mathbf{x}, \vec{\mathbf{x}}_s, \vec{f}_s) := \sum_i^{n_s} k(\mathbf{x}_{s,i}, \mathbf{x}) \cdot w_i(\vec{\mathbf{x}}_s, \vec{f}_s) \quad (6)$$

260 interpolates the given support point values  $\vec{f}_s$  at  $\vec{\mathbf{x}}_s$  for a new position  $\mathbf{x}$ . To do so, the kernel  $k$  computes the dependencies of each support point  $\mathbf{x}_{s,i}$  on the new position  $\mathbf{x}$ . Furthermore, these dependencies are weighted by  $\vec{w}(\vec{\mathbf{x}}_s, \vec{f}_s) = (w_i)$ . The weights are found by solving the linear equations system

$$f(\mathbf{x}_{s,j}, \vec{\mathbf{x}}_s, \vec{f}_s) = \sum_i^{n_s} k(\mathbf{x}_{s,i}, \mathbf{x}_{s,j}) \cdot w_i = f_{s,j} \quad \forall j = 1, \dots, n_s \quad (7)$$

such that the interpolation function fits the support points.

*Length scale.* For the RBF interpolation, a length scale  $l_{\text{rbf}}$  is required that controls the area of influence of support points. Figure 9 illustrates the interpolation for various  $l_{\text{rbf}}$ . For  $l_{\text{rbf}} \leq 1d$ , the area of influence is small and the holes are so small that they become separated. Contrary, large  $l_{\text{rbf}} \geq 3.5d$  the hole shapes are smooth and connected, but support points influence a large area, which might lead to additional artificial holes. Such artificial holes can be seen in the corners in Fig. 9 for  $l_{\text{rbf}} = 4d$ . This is a result of overshooting. Suitable values for  $l_{\text{rbf}}$  lead to smooth, connected hole shapes without additional artificial holes. As can be seen in Fig. 9, this is the case for values between  $2d$  and  $3d$ . In this work, we choose  $l_{\text{rbf}} = 2.5d$ .  
270

*Threshold.* The hole contours defined by

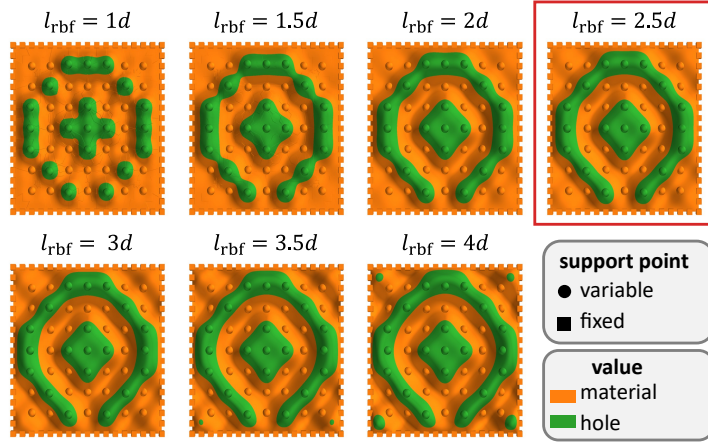


Figure 9: RBF interpolation for various length scales  $l_{\text{rbf}}$ . A length scale of  $1 d$  leads to isolated holes and a length scale of  $3.5 d$  leads to artificial holes due to the overshooting interpolation. A length scale  $l_{\text{rbf}} = 2.5d$  is used, because it can feature smooth diagonal slits and does not overshoot.

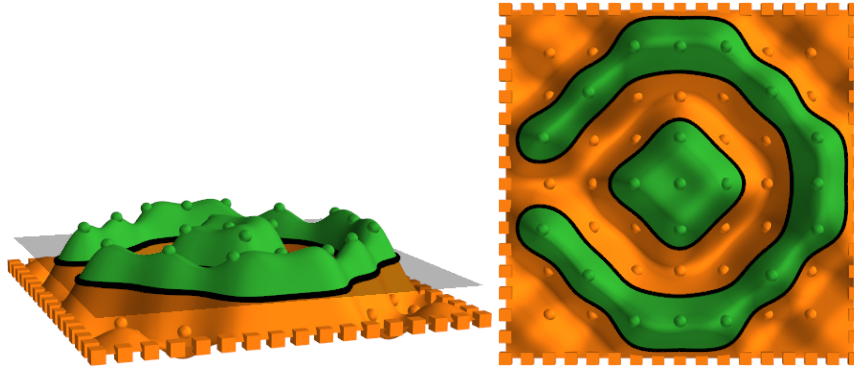


Figure 10: The hole shape (black line) is defined as the intersection of the threshold plane (grey) and the interpolation function  $f(\mathbf{x}, \dots)$  (green/orange).

$$\Gamma(\vec{f}_v) := \{\mathbf{x} \in \mathbb{X} \mid f(\mathbf{x}, \vec{\mathbf{x}}_s, \vec{f}_s(\vec{f}_v)) = f_i\} \quad (8)$$

are generated by intersecting the interpolation function  $f(\mathbf{x}, \dots)$  with a plane, as illustrated by Figure 10. This plane is placed at a threshold value of  $f_i = 0.5$ , which is halfway between the min and max levels of the support points and its intersection with  $f(\mathbf{x}, \dots)$  is computed numerically. For this, the interpolation function  $f(\mathbf{x}, \dots)$  is evaluated at  $1000 \times 1000$  points in the design space. Next, the marching squares algorithm [39], as the 2D form of the marching cubes algorithm [40], is used to extract contours  $\Gamma = \bigcup_i \Gamma_{h,i}$  as a sequence of closed poly-lines  $\Gamma_{h,i}$ . Each poly-line  $\Gamma_{h,i} = (\mathbf{x}_j, \dots)$  is a lists of points  $\mathbf{x}_j$ . A region inside a poly-line  $\Gamma_{h,i}$  is either a hole or an (invalid) material island inside a hole.

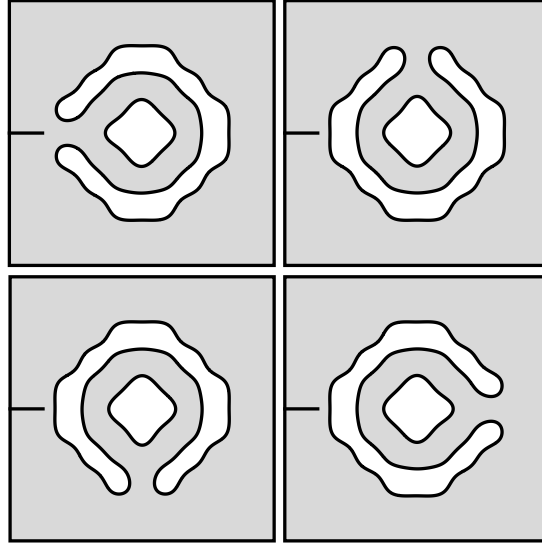


Figure 11: Cracks approaching from four sides of the hole.

*Validation.* Designs that lead to material islands are rejected, but first a criterion is needed to determine  
 280 whether a material island is present or not. The marching square algorithm returns an anti-clockwise poly-  
 line  $\Gamma_{h,i}$  if  $\Gamma_{h,i}$  encloses a hole and a clockwise poly-line  $\Gamma_{h,i}$  if  $\Gamma_{h,i}$  encloses a material island. Using this,  
 we define a validation function ( $\text{isvalid}(\Gamma)$ ), which rejects  $\Gamma$ , if  $\Gamma$  contains any clockwise poly-line  $\Gamma_{h,i}$ .

*Rotation.* We want to find valid hole patterns  $\Gamma^*$  that maximize the tensile toughness  $U_T^*$ . The tensile  
 toughness  $U_T(\Gamma)$  for holes  $\Gamma$  is evaluated as explained before. However, since a crack might approach  
 285 from various sides, the tensile toughness is evaluated for cracks approaching from four different angles  
 $\varphi \in \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$  and the minimum tensile toughness  $U_T^{\min}(\vec{f}_v)$  of all angles is computed. Instead  
 of rotating the crack for the model, the setup is kept the same and the hole contours are rotated as depicted  
 by Fig. 11. For the rotation, we define a function ( $\text{rotate}(\Gamma, \varphi)$ ) that rotates the hole contours  $\Gamma$  around the  
 center of design space  $\mathbf{x}_{\text{mid}} = \left(\frac{l_x}{2}, \frac{l_y}{2}\right)$ .

#### 2.4. Optimization for maximum tensile toughness $U_T$

290 The minimum tensile toughness  $U_T^{\min}(\vec{f}_v)$  for all rotations should be maximized. Furthermore, the  
 optimization process should be as efficient as possible and hence it is beneficial to not evaluate the same  
 geometry twice. To do so, we identify equivalent designs that can be transformed into each other either  
 by a rotation or mirror operation. If one design has already been calculated, no further equivalent designs

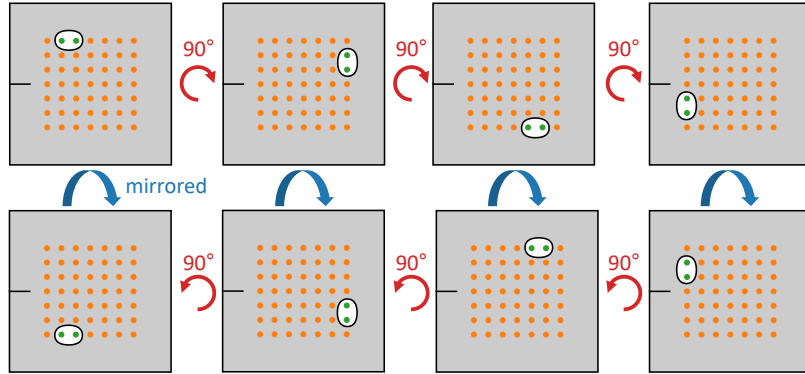


Figure 12: Equivalent designs. The variable vector  $\vec{f}_v$  of each of the shown designs yield an identical minimum tensile toughness  $U_T^{\min}(\vec{f}_v)$ .

295 need to be calculated. Figure 12 depicts such equivalent designs. The shown designs have distinct variable vectors  $\vec{f}_v$ , but since the computation of the minimum tensile toughness  $U_T^{\min}(\vec{f}_v)$  rotates the hole shape, each of those equivalent variable vectors leads to simulating all rotated designs. The outcome would be the same  $U_T^{\min}(\vec{f}_v)$ . Furthermore, designs mirrored in vertical direction are also equivalent. Since only one of those designs needs to be evaluated, the number of computations is reduced by a factor of up to 8.

#### 300 2.4.1. Optimization algorithm

As already mentioned, the minimum tensile toughness  $U_T^{\min}(\vec{f}_v)$  is the tensile toughness for a crack approaching from the worst side. We aim to maximize this minimum tensile toughness. This leads to a max-min objective

$$\begin{aligned}
 U_T^{\min}(\vec{f}_v) &:= \min_{\varphi} \{U_T(\text{rotate}(\Gamma(\vec{f}_v), \varphi))\} \\
 U_T^* &:= \max_{\vec{f}_v} \{U_T^{\min}(\vec{f}_v)\} \\
 \text{subject to : } &\begin{cases} \varphi \in \{0^\circ, 90^\circ, 180^\circ, 270^\circ\} \\ \text{isvalid}(\Gamma(\vec{f}_v)) \end{cases}
 \end{aligned} \tag{9}$$

that finds the hole pattern  $\Gamma^*(\vec{f}_v^*)$  with the maximum tensile toughness  $U_T^*$  if the crack approaches from  
 305 the worst side  $\varphi^*$ . In this work, the TBFSL algorithm [27] is used to optimize this max-min objective. The TBFSL algorithm is a trajectory-based variable neighborhood search (VNS) with two parameters:

- $d_{\text{TBFSL}}$ : the max number of simultaneously changed variables
- $r_{\text{TBFSL}}$ : the max distance of support points where variables are change

Higher values of  $d_{\text{TBFSL}}$  and  $r_{\text{TBFSL}}$  increase the probability of the TBFSL algorithm to escape local ex-  
 310 tremas. However, the computation time increases significantly with larger  $d_{\text{TBFSL}}$  and larger  $r_{\text{TBFSL}}$ .

Figure 13 illustrates the optimization process. In the first iteration  $i := 1$ , a variable vector  $\vec{f}_V^{(i)} = (0, \dots)$  of zeros is used as start vector. This is considered the currently best design  $\vec{f}_V^{(i)*} := \vec{f}_V^{(i)}$  and the corresponding currently best tensile toughness is set to  $U_T^{(i)*} := U_T^{\min}(\vec{f}_V^{(i)})$ . Next, TBFSL generates a sequence of new  
 315 variable vectors that are similar to the currently best variable vector  $\vec{f}_V^{(i)*}$ . This means, that a new variable vector contains at most  $d_{\text{TBFSL}}$  variables that differ from  $\vec{f}_V^{(i)*}$ . All those differing variables must correspond to support points that are near each other inside a rectangle with a perimeter of  $2 r_{\text{TBFSL}}$ . The generated variable vectors are put into a queue and are sorted according to the number of differing variables such that designs most similar to  $\vec{f}_V^{(i)*}$  are at the front of the queue.

320 After the queue is filled, a number of asynchronous worker processes  $n_{\text{worker}}$  is started. Each worker process picks and removes one design from the front of the queue. If an equivalent design has already been computed, the design is skipped. For all other designs, the iteration counter is incremented and the minimum tensile toughness  $U_T^{\min}(\vec{f}_V^{(i)})$  is computed. The evaluation of  $U_T^{\min}(\vec{f}_V^{(i)})$  uses a technique called  
 325 *early termination* that is described in the next section and accelerates the optimization process by considering intermediate results and stopping bad performing simulations. If  $U_T^{\min}(\vec{f}_V^{(i)})$  cannot be evaluated due to some error, the design  $\vec{f}_V^{(i)}$  is discarded. Such an error can occur in cases where a crack comes too close to a hole surface and the meshing algorithm generates distorted elements. Note, that such a failed design could be the best design. However, since the number of failed designs is negligible compared to the total number  
 330 of designs, we assume this to be unlikely.

The new minimum tensile toughness  $U_T^{\min}(\vec{f}_V^{(i)})$  is compared to the best tensile toughness  $U_T^{(i-1)*}$  of the previous step and the new best tensile toughness  $U_T^{(i)*}$  and best design  $\vec{f}_V^{(i)*}$  are updated. If the best values have changed, the queue is cleared and then filled with new designs as explained above. The optimization



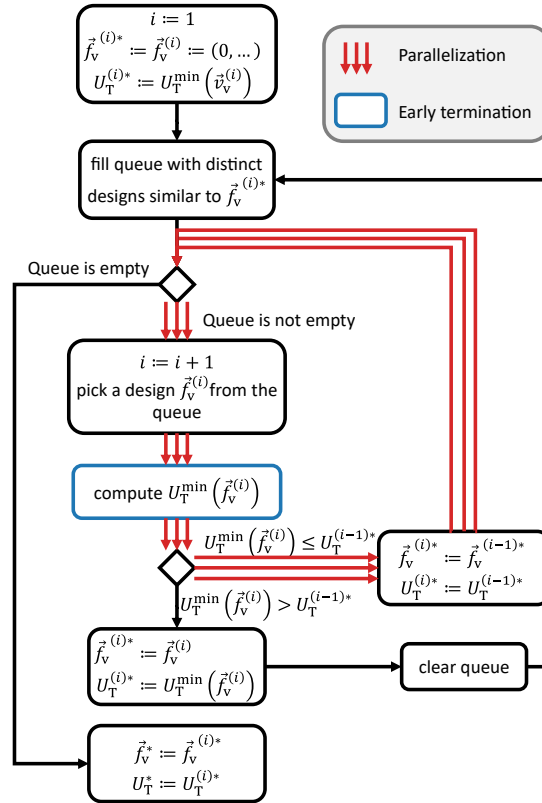


Figure 13: Flow chart of the TBFSL optimization process.

335 stops when all designs in the queue have been investigated and none of those designs from the most recent  
 340 queue have changed the best value of  $U_T^{\min}$ .

#### 2.4.2. Early termination

As already mentioned, a technique called early termination is used to accelerate the optimization process. Early termination is used in applications like hyperparameter optimization [41] to reduce the computa-  
 340 tional costs by stopping an evaluation that has bad intermediate results before completion. In this work, the  
 objective  $U_T^{\min}(\vec{f}_v)$  evaluates the tensile toughness for four rotated designs and each of those four evaluations  
 provides an intermediate result for the minimum tensile toughness  $U_T^{\min}(\vec{f}_v^{(i+1)})$ . If an intermediate result is  
 already worse than the  $U_T^{(i)*}$ , the evaluation of  $U_T^{\min}(\vec{f}_v^{(i+1)})$  is stopped and instead returns the intermediate  
 result.

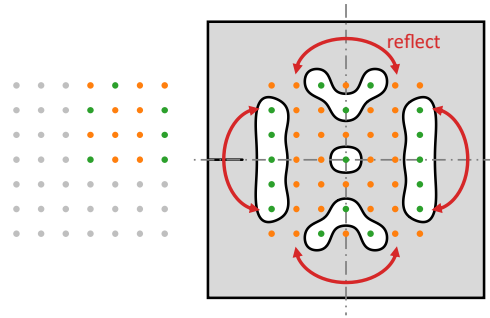


Figure 14: Illustrative example of the *reflected* pattern with variables corresponding to the variable support points in the top right quarter. The grey dot dashed lines represent the reflection plane.

### 345 2.4.3. Patterns

The optimization is performed for all variable support point variables  $n_v$ . We call this the *full* pattern. Furthermore, additional optimization runs are performed with repetition patterns that reduce the number of variables. Two patterns are considered: A *reflected* and a *rotated* pattern.

350 A *reflected* pattern shown in Fig. 14 uses only variables that correspond to the variable support points in the top right quarter. The values of the other variable support points are determined by reflecting the top right quarter around the reflection planes. The reflection planes are shown as grey dash-dotted lines. Since the *reflected* designs are the same if they are rotated by  $180^\circ$ , only two rotations ( $0^\circ$  and  $90^\circ$ ) need to be computed.

355

Figure 15 illustrates a *rotated* pattern. The values of the variable support points in the right top quadrant are optimized. By rotating the top right quarter around the center of the specimen, the values of the other variable support points are determined. Since the design stays the same for all considered rotations, only the  $0^\circ$  design is computed.

### 360 2.4.4. Optimization runs

To apply the optimization procedure described before, some parameters need to be set. The parameters used in this work are summarized here. The TBFSL optimization method requires two parameters  $d_{\text{TBFSL}}$ ,  $r_{\text{TBFSL}}$ . As discussed in Section 2.4.1, there is a trade-off between good results and a manageable computation time. Retzl et al. [27] demonstrated that the optimization is more efficient for small param-

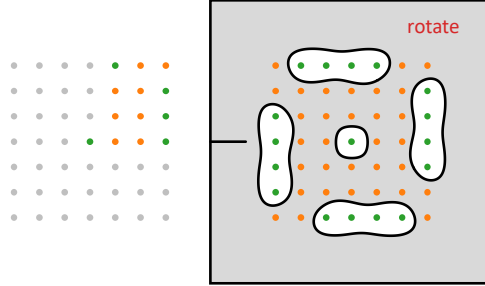


Figure 15: Illustrative example of the *rotated* pattern with variables corresponding to the variable support points in the top right quarter. The rotation is performed around the center of the specimen in three steps.

365 eters, but if parameters become too small, like  $d_{\text{TBFSL}} = 1$  or  $r_{\text{TBFSL}} = 1 \cdot 2d$ , the probability for finding the global optimum design significantly decreases. In this work, the parameters are set to  $d_{\text{TBFSL}} = 3$  and  $r_{\text{TBFSL}} = 3 \cdot 2d$ . In the example investigated by Rettl et al. [27], these parameters lead to good results and were efficient.

370 The grid size influences the radius of holes significantly and a hole contour can never cross a support point. To compensate this drawback, the grid size is varied. In total, 21 optimization runs are performed with various grid sizes and pattern types:

1. 5 runs use the *full* pattern with grid sizes between  $n_{\text{grid}} = 3$  and  $n_{\text{grid}} = 7$ .
2. 8 runs use the *reflected* pattern with grid sizes between  $n_{\text{grid}} = 3$  and  $n_{\text{grid}} = 10$ .
- 375 3. 8 runs use the *rotated* pattern with grid sizes between  $n_{\text{grid}} = 3$  and  $n_{\text{grid}} = 10$ .

The optimization runs utilize  $n_{\text{worker}} = 6$  asynchronous worker processes.

### 3. Results and discussion

Prior to the actual optimization, a crack increment length  $\Delta a$  for accurately predicting crack paths and therefore  $U_{\text{T}}$  is identified. With this  $\Delta a$ , results of some simple designs are first evaluated to show typical  $U_{\text{T}}$  values obtained. Then, optimization results for three pattern types (*full*, *reflected*, *rotated*) and for various grid sizes are presented and discussed.

380

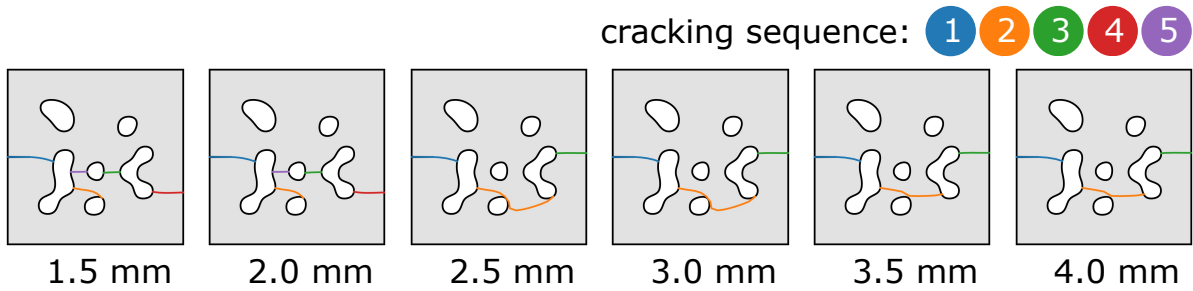


Figure 16: Crack paths depending on the crack increment length  $\Delta a$  for design C.

### 3.1. Study of $\Delta a$

From preliminary runs of the optimization algorithms, six designs with  $n_{\text{grid}} = 7$  that had found to be sensitive to  $\Delta a$  were selected. These six designs (A-F) shown in Fig. 2 are used for checking if the method  
 385 for crack propagation, crack initiation, and cracks running into holes is robust and accurate. The crack increment length  $\Delta a$  is varied between 1.5 mm and 4 mm in steps of 0.5 mm. The obtained sequence of cracks and the qualitative crack path is similar for all computed  $\Delta a$  values for all designs except for design C. Figure 16 shows the crack paths for design C for varied  $\Delta a$ . The crack path changes significantly for values  $\Delta a > 2$  mm. This change can also be seen in Fig. 17, which shows the relative tensile toughness for the designs  
 390 A-F for various  $\Delta a$ . The tensile toughness evaluated with  $\Delta a = 1$  mm serves as reference value. Above  $\Delta a = 2$  mm,  $U_T$  increases distinctively (though only by 4%) for design C. All other designs considered here are fairly accurate also for larger  $\Delta a$ . For all further computations, a crack increment length of  $\Delta a = 2$  mm is selected. For this selected  $\Delta a = 2$  mm,  $U_T$  deviates by less than 1% compared to the reference for all designs.

Figure 18 depicts the crack growth in design C with  $\Delta a = 2$  mm. Figure 18a shows the  $F(u)$  curve and the fully broken design. Figure 18b shows the critical displacements  $u_c$  for crack propagation and crack  
 400 initiation. The numbers **1-5** highlight the positions where cracks propagate (**1**) or initiate (**2-5**). At first, the plate contains only the initial crack. When the load is applied,  $F(u)$  increases with a constant slope until  $u_c$  for the crack propagation of the initial crack is met. At this point, marked as **1**, the initial crack starts to grow. The growing crack reduces the stiffness of the plate and thus the force decreases. Note, that Fig. 18b shows a slightly ascending  $u_c$  after **1**. This is caused by a stable crack propagation, where  $u_c$  gradually increases during crack growth and can be seen by a gradual drop in the  $F(u)$  curve after point

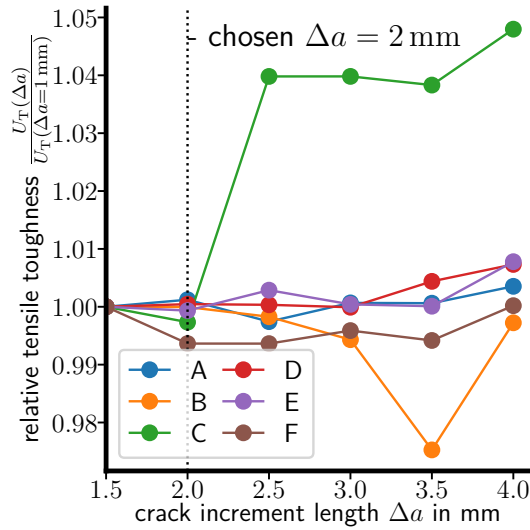
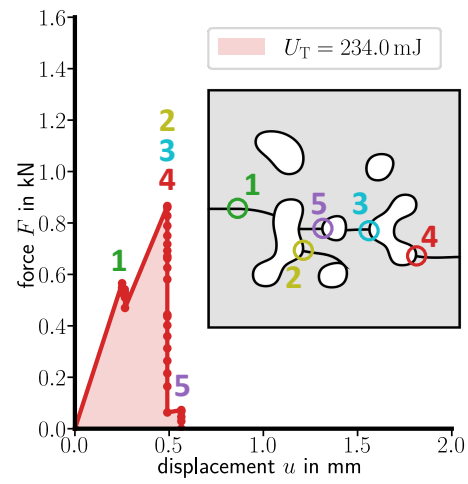
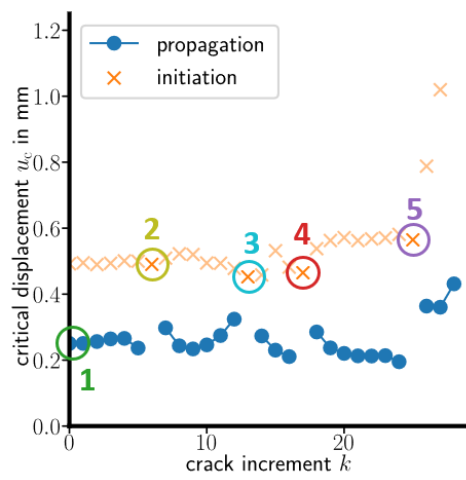


Figure 17: Convergence of the tensile toughness  $U_T$  obtained in the six designs A-F in respect to the crack increment length  $\Delta a$  used. Therefore,  $\Delta a = 2$  mm is considered sufficiently small.

1. After a few increments, the first crack hits a hole and is deactivated. Then, a new crack initiates. Here, the local stresses play a crucial role, which are determined by the local curvature radius of the hole as well as other holes nearby that may relax some of the stresses. The new crack initiates, as marked by **2**, once a displacement of 0.5 mm is applied. According to Fig. 18b,  $u_c$  for propagating the new crack directly after **2** is at about 0.3 mm which is lower than the displacement of 0.5 mm applied at **2**. Consequently, the crack growth is unstable until it hits the bottom circular hole. The critical displacements for the next two initiations and propagation events **3** and **4** are still smaller than the displacement for **2**. This means that the cracks **3** and **4** also initiate immediately with unstable crack growth after **2**. This can be seen in Fig. 18a, where all three events **2**, **3**, and **4** occur at the same applied displacement. The final crack **5** initiates at a higher displacement of about 0.55 mm, but since the plate is almost fully broken at that point, the remaining stiffness is low. This leads to a small force for the initiation below 100 N, and the area underneath  $F(u)$  from **4** to **5** remains small and thus contributes only very little to the tensile toughness  $U_T$ .

### 3.2. $F(u)$ curves for simple setups

To give a reference for the following optimization results, some simple hole designs using a  $7 \times 7$  grid are presented in Fig. 19. We present a plate without any holes (Fig. 19a), one with all  $7 \times 7$  voxels empty (Fig. 19b), one with a single diamond-shaped hole (Fig. 19c), one with a small hole at the center (Fig. 19d)

(a)  $F(u)$  curve(b)  $u_c$  curveFigure 18: Results for test design C with a crack increment length  $\Delta a = 2$  mm.

and two designs with a  $1 \times 7$  elongated hole at the center with vertical (Fig. 19e) and horizontal (Fig. 19f) orientation.

The  $F(u)$  curves, depicted in Fig. 19, first increase linearly until a critical displacement, where  $F$  drops. This drop in  $F$  is associated with the propagation of the initial crack and the associated stiffness reduction. The more the crack propagates in this step, the bigger the drop in  $F$ . If this first drop has not been fatal for the plate (as for the plate without holes in Fig. 19a), the force then increases up to a higher critical displacement with the reduced stiffness, where  $F$  drops again. Contrary to design C in Fig. 18, this second drop leads to ultimate failure of these simple plate designs, which means that the initiated crack propagates through the whole remaining cross-section of the plate. The critical displacement at the second drop depends on the local stresses in the region of crack initiation, such that for a large curvature radius at the initiation point (designs in Fig. 19b,e) the maximum displacement is about 1 mm, whereas for a smaller radius (designs in Fig. 19c,d,f) it is much lower with about 0.4 mm.

To obtain a large  $U_T$ , both the critical displacements for crack growth and crack initiation should be large, but at the same time, the stiffness should be large: this can clearly be seen when comparing the design with all voxels empty ( $U_T = 337$  mJ, see Fig. 19b) to the design with the vertical hole ( $U_T = 759$  mJ, see Fig. 19e), which have similar  $u_c$  for crack growth and crack initiation, but a significantly different stiffness. The highest  $U_T$  can therefore be obtained if the holes and the growth of the initial crack into the closest hole reduce the plate stiffness as little as possible and the stress at the following initiation position is low. The stresses can be reduced by a large curvature radius at the initiation position or by additional holes that relieve stresses at the initiation position. The vertical hole could be moved closer to the initial crack to reduce the stiffness drop due to the initial crack growth, but apart from that, the design with the vertical hole is already ideal for the given crack. This mimics the effect of a layered structure loaded in tension. For other crack locations, however, this design can be much worse in terms of its  $U_T$ , as shown for the horizontal hole in Fig. 19f. This plate has a low initial stiffness, a big drop in the stiffness, and a low  $u_c$  for crack initiation, resulting in  $U_T = 90.7$  mJ, an even lower value than that of the full plate ( $U_T = 109$  mJ). This indicates that there is no simple and trivial solution to the optimization problem in this work.

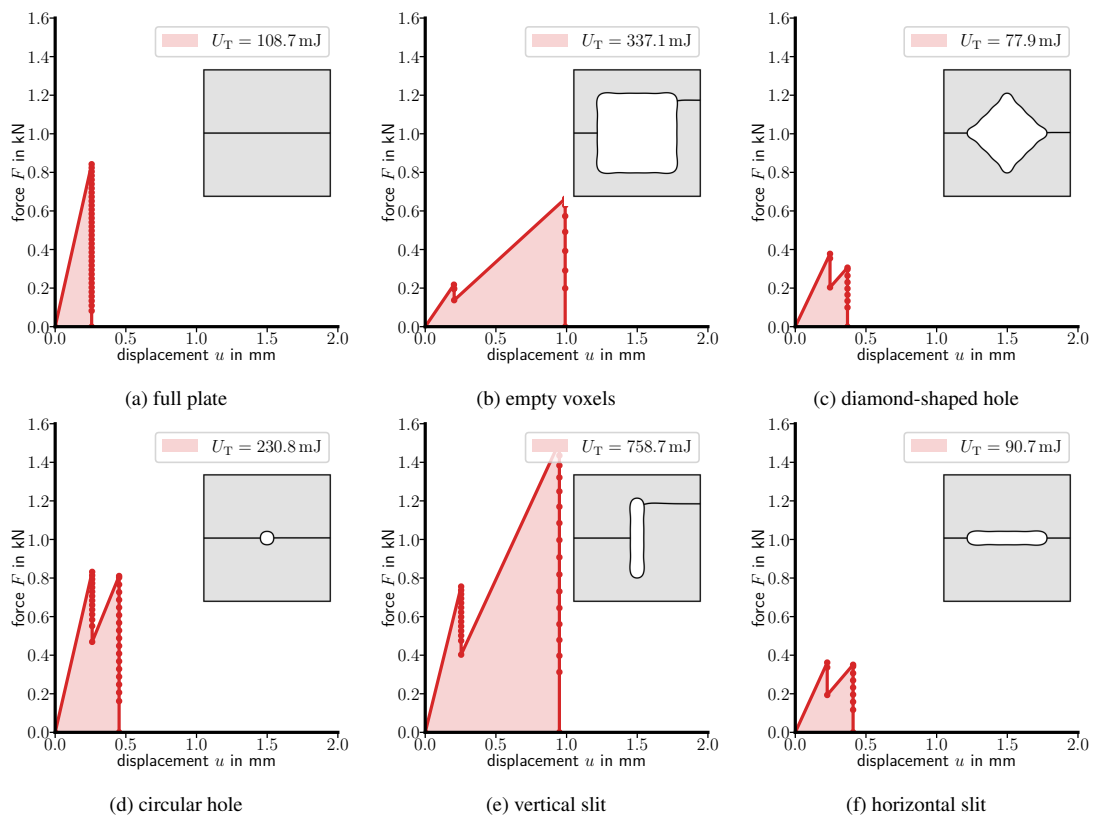


Figure 19: Resulting  $F(u)$  curves and crack paths for some simple hole designs. For the full plate without holes,  $U_T = 109 \text{ mJ}$ , and the highest  $U_T = 757 \text{ mJ}$  is reached for the vertical slit, which only has  $U_T = 91 \text{ mJ}$  if rotated by  $90^\circ$ .



### 3.3. Optimizing the hole geometry

As defined in Sec. 2.4.4, geometries are optimized using the three pattern types *full*, *reflected*, and *rotated*. The size of the  $n_{\text{grid}} \times n_{\text{grid}}$  grid varies between  $n_{\text{grid}} = 3$  and  $n_{\text{grid}} = 10$ . The optimization with the *full* pattern is only performed up to a grid size of  $n_{\text{grid}} = 7$ , because this optimization run already took more than 52 hours. The computation time and the optimization progress for  $U_T$  are shown in Fig. 20. The detailed results of all optimization runs are provided in the supplementary materials of this work. Table 1 provides a summary of all runs, including the number of failed iterations. Each iteration consists of up to four model evaluations (for the rotation angles) of each design. Each model evaluation performs on average 26 FEM simulations for the crack propagation and initiation. The total number of FEM computations for all optimizations runs is more than 280,000. For such a large number of FEM models, it is not feasible to completely eliminate failed computations. Only 394 (0.14%) of more than 280,000 computations were not computed successfully. The failed iterations, which were less than 5% of the total iterations, were found to be caused by failed Abaqus meshing for unclear reasons and cases where a crack close to a hole surface led to distorted elements. Both cases occur more often with finer grids, where the meshing parameters and the  $\Delta a$  selected may have been too large, since it was fixed at 2 mm independently on  $n_{\text{grid}}$ .

Figure 20 shows that the *reflected* pattern leads to the best designs with the highest  $U_T^*$ . There is a weak correlation for larger grids resulting in better designs. Considering the logarithmic scale of the computation time in Fig. 20, it can be seen that optimization runs on larger grids take much longer. Although the *full* pattern has the highest computation time, it performs the worst. On average, the *rotated* and *reflected* patterns find better  $U_T^*$  than the *full* pattern in less time on the same grid size. It is obvious that the *full* pattern needs more computation time, because for the same grid size, it has more variables and needs to perform model evaluation for cracks from all four directions. The *rotated* pattern needs to consider only one crack direction, because it is rotational symmetric. The *reflected* pattern models evaluations for cracks from two directions.

However, the better  $U_T^*$  of the *rotated* and *reflected* patterns is counter-intuitive, because the set of possible designs is greater for the *full* pattern and also contains all possible designs of the other patterns.

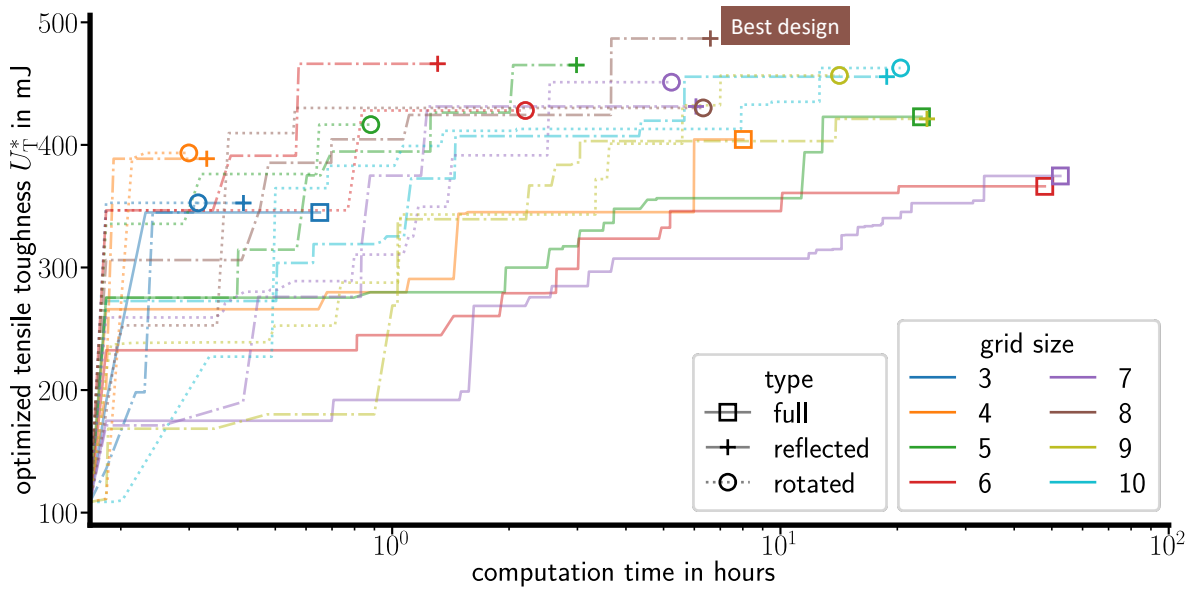


Figure 20: The optimized tensile toughness  $U_T^*$  depending on the computation time of the optimization. The value  $U_T^*$  increases with the progress in the optimization. Optimizations are carried out for the three types of patterns *full*, *reflected*, and *rotated* for grid sizes between 3 and 10. The highest  $U_T^*$  is found for the *reflected* pattern and a grid size of 8.

475 Consequently, the *full* pattern should be able to find the designs optimized by the *rotated* and *reflected* patterns. This is not the case, however, since it gets stuck in local optima. If the *rotated* and *reflected* patterns fall in the same local optima, they might escape it, because a change in one variable changes up to four support point values, due to the rotation and reflection, respectively. This leads to more diverse designs and higher probability of escaping local optima.

480

In the following, the designs and detailed results of five selected optimizations are presented. This is done for the *full* pattern with a  $7 \times 7$  grid, the *reflected* pattern with a  $5 \times 5$  grid and a  $8 \times 8$  grid (which has the highest  $U_T^*$ ), and the *rotated* pattern with a  $4 \times 4$  grid and a  $10 \times 10$  grid.

### 485 3.3.1. Best overall design

Figure 21 shows the best design found of a *reflected* pattern and  $n_{\text{grid}} = 8$  with  $U_T^* = 486.8$  mJ. The initial crack starts to grow at a force of about 500 N and is then stopped after a few increments by a small vertical slit. The effect of this crack on the stiffness is quite small and the force drops only by a small

Table 1: Results of all optimization runs on  $n_{\text{grid}} \times n_{\text{grid}}$  grids with the patterns *full*, *reflected*, and *rotated* that reduce the number of variables  $n_{\text{var}}$ .

pattern	$n_{\text{grid}}$	$n_{\text{var}}$	$U_T^*$ in mJ	iterations	failed iterations	model evaluations	computation time in hh:mm
<i>full</i>	3	9	344.9	24	0	45	00:39
	4	16	404.4	397	0	506	08:01
	5	25	422.9	1038	1	1352	23:04
	6	36	366.2	1548	4	2149	47:57
	7	49	374.7	1619	3	2136	52:34
<i>reflected</i>	3	4	352.6	10	0	18	00:24
	4	4	388.7	10	0	17	00:19
	5	9	465.1	122	0	160	02:59
	6	9	466.2	55	0	72	01:18
	7	16	431.4	255	4	321	06:03
	8	16	486.8	235	0	277	06:35
	9	25	421.2	850	8	949	23:50
10	25	455.6	644	8	692	18:47	
<i>rotated</i>	3	3	352.7	6	0	6	00:18
	4	4	393.4	10	0	10	00:17
	5	7	416.5	52	0	52	00:52
	6	9	428.1	99	1	99	02:12
	7	13	451.2	251	5	251	05:14
	8	16	430.1	259	6	259	06:19
	9	21	456.7	602	15	602	14:11
	10	25	462.7	687	35	687	20:24

amount. Next, the crack initiates at the vertical slit. However, the holes above and below the vertical slit  
490 act as stress relief notches [42] by lowering the stresses by shielding. Consequently, a high force of almost  
1.4 kN is required for the crack initiation. This shielding and stress relief mechanism is the main reason for  
the high  $U_T$ . The crack then grows further into the hole at the center, which is again shielded by stress relief  
holes above and below. Although the stresses around the central hole are reduced by the stress relief holes,  
the force for the initiation is lower than before at about 1000 N. This is visible as the small third peak in the  
495  $F(u)$  curve. Finally, the crack hits the right vertical slit, but immediately initiates again and grows all the  
way through to the right edge of the plate.

Although the *reflected* pattern is not constrained to rotational symmetric designs, the optimized design  
is also rotational symmetric. One might expect that the *rotated* pattern evaluated for the same grid finds the  
500 same design. However, it found a worse design, because the TBFS optimization usually does not find the

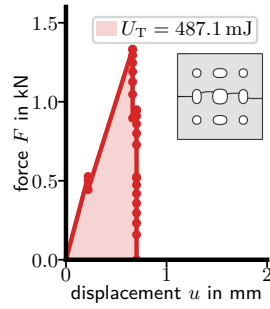


Figure 21: Best design from the optimization runs with the *reflected* pattern with  $n_{\text{grid}} = 8$ .

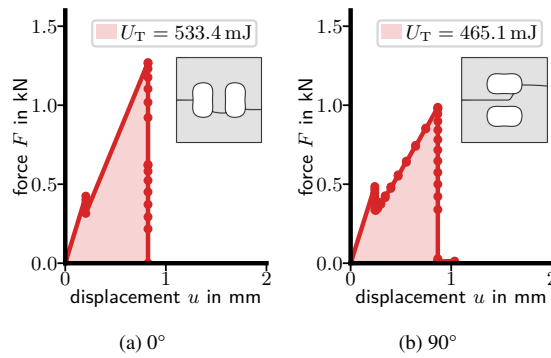


Figure 22: optimized design found using the *reflected* pattern with  $n_{\text{grid}} = 5$ .

best design, unless higher values for parameters  $d_{\text{TBFS}}$  and  $r_{\text{TBFS}}$  are chosen.

### 3.3.2. Reflected pattern design with $n_{\text{grid}} = 5$

Another interesting design was found using the *reflected* pattern with  $n_{\text{grid}} = 5$ . Figure 22 shows this design, which consists of two vertical slits in the  $0^\circ$  rotation state (Fig. 22a) and two horizontal slits in the  $90^\circ$  rotation state (Fig. 22b). For the  $0^\circ$  rotation, the initial crack grows into the left vertical slit. This slit behaves similar to the vertical slit presented in Fig. 19e. Since this slit has a large curvature radius, the notch effect and thus the stress concentration are small. This leads to a high critical force of about 1.3 kN. Furthermore, the big slits reduce the stiffness more than in the previous design shown in Fig. 21. Consequently, for the same force, a higher displacement is necessary. The combination of the high displacement and the high force result in a high  $U_T$  of 533.4 mJ.

However, if the design is rotated by  $90^\circ$  (Fig. 22b),  $U_T$  decreases. First, the crack grows midway between the two horizontal slits. The two slits shield the crack and there is almost no load at the crack tip.

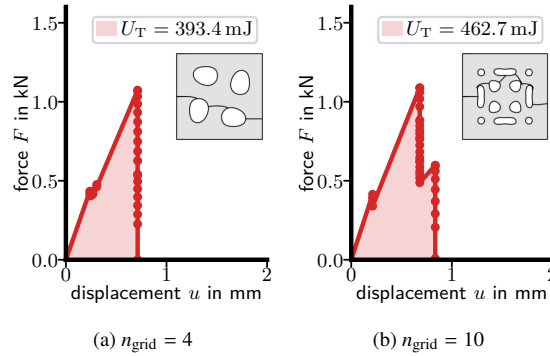


Figure 23: Two optimized designs found using the *rotated* pattern and various grid sizes.

Therefore, stable crack growth occurs, which is indicated by dots in the  $F(u)$  plot during the increase of the  
 515 force towards the second peak. At the second peak, the crack stops prior to hitting a hole approximately at  
 the center of the plate, because the crack tip is shielded by the two holes. At this stage, the critical force for  
 the initiation of a new crack is lower. Next, a new crack initiates at the right side of the upper horizontal slit  
 and immediately grows into the right side of the plate. The plate is now almost broken and has a very low  
 stiffness. Only then, the first crack finally grows into the upper hole.

### 520 3.3.3. *Rotated pattern designs*

Figure 23 shows two designs that were optimized using the *rotated* pattern both on a coarse grid  $n_{\text{grid}} = 4$   
 (Fig. 23a) and a fine grid  $n_{\text{grid}} = 10$  (Fig. 23b). The coarse grid has in total 4 variables and can only represent  
 very simple designs. The optimized coarse grid design consists of four oval holes with  $U_T = 393.4$  mJ. The  
 crack grows from left to the right and hits two of the holes. For the fine grid with its 25 variables, the best  
 525 design found ( $U_T = 462.7$ ) is much more complex. In total, the plate contains 11 holes of various shapes.  
 Like in the best overall design, shown in Fig. 21, the crack starts here at about 500 N and runs then into a  
 vertical slit with stress relief holes above and below. A high load is required to initiate new cracks. In the  
 center, there are four circular holes that deflect the crack further. The final crack initiates at a force of about  
 600 N at the right slit and propagates to the right edge of the plate.

### 530 3.3.4. *Full pattern design with $n_{\text{grid}} = 7$*

The optimization using the *full* pattern on a fine grid with  $n_{\text{grid}} = 7$  involves 49 binary variables and took  
 more than 52 hours to compute. About  $2^{49} \approx 5 \cdot 10^{14}$  variable combinations would be possible, of which

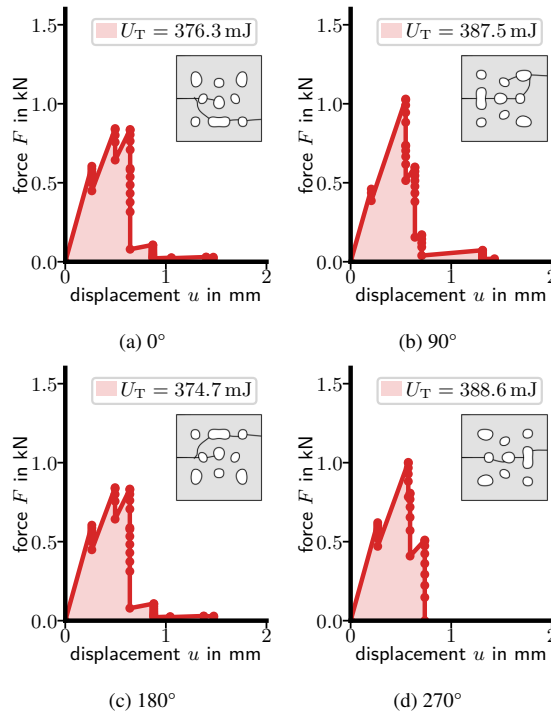


Figure 24: Design optimized by the *full* pattern with  $n_{\text{grid}} = 7$ .

only 1619 were considered during the optimization. Figure 24 shows the  $F(u)$  curves of all four rotations of the optimized *full*  $7 \times 7$  design. The plate has no symmetry and contains 9 holes without any obvious arrangement. The holes significantly deflect the crack and lead to many initiations. This is indicated by the number of peaks in the  $F(u)$  curves that shows 5, 6, 5, and 3 peaks for the  $0^\circ$ ,  $90^\circ$ ,  $180^\circ$ , and  $270^\circ$  rotations, respectively. Although the  $U_T$  of these designs (minimum  $U_T$  of 374.7 mJ) is lower compared to the best design found for the *reflected*  $8 \times 8$  grid, the displacement for ultimate failure with more than 1.5 mm is much higher for all rotations except for  $270^\circ$ . This is due to small remaining material brides that have a low stiffness and only contribute slightly to  $U_T$ .

## Conclusions

In this work, an optimization strategy was developed and implemented based on an optimization method presented by Retzl [27]. The optimization strategy was applied to an example to investigate the influence of multiple holes on crack propagation and the toughness of a specimen. For the example, arbitrary shaped holes in a 2D plate were optimized to maximize the tensile toughness of the quadratic plate. The holes were

represented as contour lines of a parameterized function whose parameters act as optimization variables. Crack initiations were predicted by Taylor's criterion and brittle crack propagation was predicted using configurational forces.

550 Optimization runs that reduced the variables with symmetry planes resulted in reduced computation time but also in improved designs. The optimal design, which involved 16 binary variables and included both x- and y-symmetry planes, achieved a tensile toughness of 486.8 mJ in the 80 mm × 80 mm plate with PMMA properties. The optimization run to obtain this design took 6:35 h to compute on a 6 core 3 GHz CPU. Notably, this design exhibited 4.5 times higher tensile toughness than a solid plate and 2.1  
555 times higher tensile toughness than a plate with one circular hole. Additionally, the strategy of using a hole shielded by stress relief holes to stop crack propagation appears beneficial. This approach decreases stresses at the hole surface, which leads to higher forces for the crack initiation, and thus increases the tensile toughness. Furthermore, the high tensile toughness improves, if the plate stiffness remains high despite the propagating cracks.

#### 560 **CRedit authorship contribution statement**

**Matthias Rettl:** Conceptualisation, Methodology, Software, Investigation, Visualisation, Writing — original draft preparation, Writing — review and editing. **Martin Pletz:** Conceptualisation, Methodology, Software, Investigation, Visualisation, Writing — original draft preparation, Writing — review and editing. **Clara Schuecker:** Conceptualisation, Writing — review and editing.

#### 565 **Declaration of competing interest**

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### **Data availability**

Data will be made available on request.

570 **References**

- [1] R. O. Ritchie, The conflicts between strength and toughness, *Nature Materials* 10 (2011) 817–822. doi:10.1038/nmat3115.
- [2] M. Ahmadi, K. Ehrmann, T. Koch, R. Liska, J. Stampfl, From Unregulated Networks to Designed Microstructures: Introducing Heterogeneity at Different Length Scales in Photopolymers for Additive Manufacturing, *Chemical Reviews* 124 (2024) 3978–4020. doi:10.1021/acs.chemrev.3c00570.
- 575 [3] P. Fratzl, O. Kolednik, F. D. Fischer, M. N. Dean, The mechanics of tessellations – bioinspired strategies for fracture resistance, *Chemical Society Reviews* 45 (2016) 252–267. doi:10.1039/C5CS00598A.
- [4] R. Bermejo, R. Daniel, C. Schuecker, O. Paris, R. Danzer, C. Mitterer, Hierarchical Architectures to Enhance Structural and Functional Properties of Brittle Materials, *Advanced Engineering Materials* 19 (2017) 1600683. doi:10.1002/adem.201600683.
- 580 [5] M. Grossman, D. Pivovarov, F. Bouville, C. Dransfeld, K. Masania, A. R. Studart, Hierarchical Toughening of Nacre-Like Composites, *Advanced Functional Materials* 29 (2019) 1806800. doi:10.1002/adfm.201806800.
- [6] G. X. Gu, F. Libonati, S. D. Wettermark, M. J. Buehler, Printing nature: Unraveling the role of nacre’s mineral bridges, *Journal of the Mechanical Behavior of Biomedical Materials* 76 (2017) 135–144. doi:10.1016/j.jmbbm.2017.05.007.
- [7] M. A. Monn, J. C. Weaver, T. Zhang, J. Aizenberg, H. Kesari, New functional insights into the internal architecture of the laminated anchor spicules of *Euplectella aspergillum*, *Proceedings of the National Academy of Sciences* 112 (2015) 4976–4981. doi:10.1073/pnas.1415502112.
- 585 [8] M. E. Launey, M. J. Buehler, R. O. Ritchie, On the Mechanistic Origins of Toughness in Bone, *Annual Review of Materials Research* 40 (2010) 25–53. doi:10.1146/annurev-matsci-070909-104427.
- [9] A. Parvizi, K. W. Garrett, J. E. Bailey, Constrained cracking in glass fibre-reinforced epoxy cross-ply laminates, *Journal of Materials Science* 13 (1978) 195–201. doi:10.1007/BF00739291.
- 590 [10] O. Kolednik, J. Predan, F. D. Fischer, P. Fratzl, Bioinspired Design Criteria for Damage-Resistant Materials with Periodically Varying Microstructure, *Advanced Functional Materials* 21 (2011) 3634–3641. doi:10.1002/adfm.201100443.
- [11] Y. Beygelzimer, R. Kulagin, P. Fratzl, Y. Estrin, The Earth’s Lithosphere Inspires Materials Design, *Advanced Materials* 33 (2021) 2005473. doi:10.1002/adma.202005473.
- 595 [12] G. H. Michler, F. J. Baltá-Calleja, *Nano- and Micromechanics of Polymers: Structure Modification and Improvement of Properties*, Hanser Publications, Cincinnati, 2012.
- [13] Z. Jia, L. Wang, 3D printing of biomimetic composites with improved fracture toughness, *Acta Materialia* 173 (2019) 61–73. doi:10.1016/j.actamat.2019.04.052.
- [14] M. Sistaninia, O. Kolednik, Improving strength and toughness of materials by utilizing spatial variations of the yield stress, *Acta Materialia* 122 (2017) 207–219. doi:10.1016/j.actamat.2016.09.044.
- 600 [15] A. Tiwari, J. Wiener, F. Arbeiter, G. Pinter, O. Kolednik, Application of the material inhomogeneity effect for the improvement of fracture toughness of a brittle polymer, *Engineering Fracture Mechanics* 224 (2020) 106776. doi:10.1016/j.engfracmech.2019.106776.
- [16] R. Bermejo, “Toward seashells under stress”: Bioinspired concepts to design tough layered ceramic composites, *Journal of the European Ceramic Society* 37 (2017) 3823–3839. doi:10.1016/j.jeurceramsoc.2017.04.041.
- 605 [17] J. Zechner, O. Kolednik, Paper multilayer with a fracture toughness of steel, *J Mater Sci* (2013).
- [18] A. J. Kinloch, S. J. Shaw, D. A. Tod, Deformation and fracture behaviour of a rubber-toughened epoxy: 1. Microstructure and fracture studies, *Polymer* 24 (1983) 1341–1354. doi:10.1016/0032-3861(83)90070-8.
- [19] G. Davy, S. Hashemi, A. Kinloch, The fracture of a rubber-modified epoxy polymer containing through-thickness and surface cracks, *International Journal of Adhesion and Adhesives* 9 (1989) 69–76. doi:10.1016/0143-7496(89)90027-4.
- 610 [20] D. Yadav, T. More, B. N. Jaya, Morse-Code inspired architectures for tunable damage tolerance in brittle material systems, *Journal of Materials Research* 37 (2022) 1201–1215. doi:10.1557/s43578-022-00520-6.
- [21] D. Brescakovic, M. Kegl, O. Kolednik, Interaction of crack and hole: Effects on crack trajectory, crack driving force and fracture toughness, *International Journal of Fracture* 236 (2022) 33–57. doi:10.1007/s10704-021-00611-1.
- 615 [22] D. Brescakovic, O. Kolednik, Fracture toughness improvement due to crack deflection and crack trapping by elliptical voids or particles, *International Journal of Solids and Structures* 285 (2023) 112551. doi:10.1016/j.ijsolstr.2023.112551.
- [23] Abaqus 2020, Dassault Systèmes (2020).
- [24] M. Pletz, S. M. Frankl, C. Schuecker, Evaluation of existing and introduction of new incremental crack propagation approaches in FEM, *Theoretical and Applied Fracture Mechanics* 131 (2024) 104452. doi:10.1016/j.tafmec.2024.104452.
- 620 [25] D. Taylor, The theory of critical distances, *Engineering Fracture Mechanics* 75 (2008) 1696–1705. doi:10.1016/j.engfracmech.2007.04.007.
- [26] M. Aranda, I. García, A. Quintanas-Corominas, J. Reinoso, Crack impinging on a curved weak interface: Penetration or deflection?, *Journal of the Mechanics and Physics of Solids* 178 (2023) 105326. doi:10.1016/j.jmps.2023.105326.
- 625 [27] M. Rettl, M. Pletz, C. Schuecker, Evaluation of combinatorial algorithms for optimizing highly nonlinear structural problems, *Materials & Design* 230 (2023) 111958. doi:10.1016/j.matdes.2023.111958.



- 
- [28] A. Sapora, A. Torabi, S. Etesam, P. Cornetti, Finite Fracture Mechanics crack initiation from a circular hole, *Fatigue & Fracture of Engineering Materials & Structures* 41 (2018) 1627–1636. doi:10.1111/ffe.12801.
- [29] S. M. Frankl, M. Pletz, C. Schuecker, Improved concept for iterative crack propagation using configurational forces for targeted angle correction, *Engineering Fracture Mechanics* 266 (2022) 108403. doi:10.1016/j.engfracmech.2022.108403.
- 630 [30] M. Rettl, S. M. Frankl, M. Pletz, M. Tauscher, C. Schuecker, ConForce: Computation of configurational forces for FEM results, *SoftwareX* 26 (2024) 101718. doi:10.1016/j.softx.2024.101718.
- [31] D. Leguillon, Strength or toughness? A criterion for crack onset at a notch, *European Journal of Mechanics - A/Solids* 21 (2002) 61–72. doi:10.1016/S0997-7538(01)01184-6.
- 635 [32] M. Rettl, M. Pletz, C. Schuecker, Efficient prediction of crack initiation from arbitrary 2D notches, *Theoretical and Applied Fracture Mechanics* 119 (2022) 103376. doi:10.1016/j.tafmec.2022.103376.
- [33] J. Zheng, S. Zhu, F. Soleymani, A new efficient parametric level set method based on radial basis function-finite difference for structural topology optimization, *Computers & Structures* 297 (2024) 107364. doi:10.1016/j.compstruc.2024.107364.
- [34] G. Pingen, M. Waidmann, A. Evgrafov, K. Maute, A parametric level-set approach for topology optimization of flow domains, *Structural and Multidisciplinary Optimization* 41 (2010) 117–131. doi:10.1007/s00158-009-0405-1.
- 640 [35] Y. Estrin, Y. Bréchet, J. Dunlop, P. Fratzl (Eds.), *Architected Materials in Nature and Engineering: Archimats*, volume 282 of *Springer Series in Materials Science*, Springer International Publishing, Cham, 2019. doi:10.1007/978-3-030-11942-3.
- [36] R. L. Hardy, Multiquadric equations of topography and other irregular surfaces, *Journal of Geophysical Research* 76 (1971) 1905–1915. doi:10.1029/JB076i008p01905.
- 645 [37] M. Kanagawa, P. Hennig, D. Sejdinovic, B. K. Sriperumbudur, Gaussian Processes and Kernel Methods: A Review on Connections and Equivalences, <http://arxiv.org/abs/1807.02582>, 2018. arXiv:1807.02582.
- [38] D. B. McDonald, W. J. Grantham, W. L. Tabor, M. J. Murphy, Global and local optimization using radial basis function response surface models, *Applied Mathematical Modelling* 31 (2007) 2095–2110. doi:10.1016/j.apm.2006.08.008.
- [39] S. Van Der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, scikit-image: image processing in Python, *PeerJ* 2 (2014) e453. doi:10.7717/peerj.453.
- 650 [40] W. E. Lorensen, H. E. Cline, Marching cubes: A high resolution 3D surface construction algorithm, in: *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, 1987, pp. 163–169. doi:10.1145/37401.37422.
- [41] D. Marinov, D. Karapetyan, Hyperparameter Optimisation with Early Termination of Poor Performers, <http://arxiv.org/abs/1907.08651>, 2019. arXiv:1907.08651.
- 655 [42] M. Bijak-Zochowski, A. M. Waas, W. J. Anderson, C. E. Miniatt, Reduction of contact stress by use of relief notches, *Experimental Mechanics* 31 (1991) 271–275. doi:10.1007/BF02326071.