

# Multi-Level $hp$ -Adaptivity for Cohesive Fracture Modeling

Nils Zander<sup>\*1</sup>, Martin Ruess<sup>†3</sup>, Tino Bog<sup>1</sup>, Stefan Kollmannsberger<sup>1</sup>, and Ernst Rank<sup>1,2</sup>

<sup>1</sup>Chair for Computation in Engineering, Technische Universität München, Arcisstr. 21, 80333 München, Germany

<sup>2</sup>Institute for Advanced Study, Technische Universität München, Lichtenbergstr. 2a, 85748 Garching, Germany

<sup>3</sup>School of Engineering, University of Glasgow, Oakfield Avenue G12 8LT, United Kingdom

---

## Abstract

Discretization induced oscillations in the load-displacement curve are a well known problem for simulations of cohesive crack growth with finite elements. The problem results from an insufficient resolution of the complex stress state within the cohesive zone ahead of the crack tip. This work demonstrates that the  $hp$ -version of the finite element method is ideally suited to resolve this complex and localized solution characteristic with high accuracy and low computational effort. To this end, we formulate a local and hierarchic mesh refinement scheme that follows *dynamically* the propagating crack tip. In this way, the usually applied *static* a priori mesh refinement along the complete potential crack path is avoided, which significantly reduces the size of the numerical problem. Studying systematically the influence of  $h$ -,  $p$ - and  $hp$ -refinement, we demonstrate why the suggested  $hp$ -formulation allows to capture accurately the complex stress state at the crack front preventing artificial snap-through and snap-back effects. This allows to decrease significantly the number of degrees of freedom and the simulation runtime. Furthermore, we show that by combining this idea with the finite cell method, the crack propagation within complex domains can be simulated efficiently without resolving the geometry by the mesh.

*Keywords:* cohesive fracture, automatic  $hp$ -adaptivity, arbitrary hanging nodes, dynamic meshes, finite cell method

---

## 1 Introduction

In various engineering and industrial applications, the accurate prediction of crack initiation and propagation is of major importance to prevent catastrophic failure. Often, the fracture mechanism is not brittle along a sharp interface but occurs within a fracture process zone, in which the material gradually loses its strength.

A common approach to capture this kind of failure is the use of cohesive zone models first introduced in [1] and [2]. The basic idea of these models is to describe the cohesive forces in the fracture process zone with respect to the crack opening using a traction-separation law. In [3], this modeling strategy was combined with the finite element method to simulate crack initiation and propagation numerically. Over the decades, this approach has proven its potential in diverse fields of application

---

\*nils.zander@tum.de, Corresponding Author

†martin.ruess@glasgow.ac.uk, Corresponding Author

such as void nucleation [4], quasi-static crack growth [5–7], and e.g. dynamic crack growth [8–10]. In particular, in cases where the potential crack path is known in advance as in e.g. laminated structures, cohesive zone models yield their full potential since they can be incorporated easily in the finite element formulation [11–14]. A review about the basic principles of the method is given e.g. in [15].

The use of cohesive zone models is numerically challenging due to an “intrinsic discretization sensitivity” [16]. In particular, the cohesive zone ahead of the crack requires sufficiently small elements to prevent artificial, discretization induced oscillations in the load-displacement curve as reported in numerous works [16–29].

Studies presented in [20–23, 29] show that the cohesive zone must be resolved with at least two to five elements with linear shape functions to achieve reasonable results. Resolving the whole computational domain with such a fine resolution is prohibitively expensive. Hence, the mesh is typically refined locally along the crack path, given that it is known *a priori*. The task of further improving the discretization performance is still a topic of active research. Recently, B-spline and NURBS basis functions have been used to resolve the progressing crack with high accuracy [30–33]. The positive effect of higher-order shape functions on the artificial, discretization induced oscillations is studied in [27]. In [28], the approach has been further extended by using T-splines to refine the discretization non-uniformly along the crack path.

However, the possible crack paths can become large and spoil an efficient analysis. Laminated wing structures, for example, have dimensions of several meters whereas a high approximation accuracy is needed only in the vicinity of the cohesive failure zone of a few millimeters. Consequently, an *a priori* refinement along the full crack path adds a large number of unnecessary additional degrees of freedom to the discrete system.

To avoid this problem, alternative approaches have been developed allowing for a locally bounded refinement zone that follows the crack tip. Common strategies use adaptive hierarchical enrichment [19, 24], self-adaptive finite elements [16, 26], adaptive *h*-refinement [34], or error-driven adaptive re-meshing [35]. All these approaches significantly increase the accuracy of the approximation at the crack tip. However, they are restricted to an approximation with linear shape functions. This demands for a fine base mesh to capture the global deformation accurately.

A discretization scheme that has proven its superior approximation properties for problems with local solution features such as stress concentrations or even singularities is the *hp*-version of the finite element method [36–39]. The essential idea of this method is to use coarse, high-order finite elements to capture the large-scale solution efficiently, whereas local stress concentrations are resolved by refining the mesh towards the singular points in a geometric progression. To this end, the coarse elements are *replaced* by a set of finer elements. As the rest of the mesh remains unchanged, this approach allows to keep the refinement local and focused and no global re-meshing is required. The high computational performance of this approach has been demonstrated in the context of various applications [40–45].

The high approximation quality of *hp*-finite elements seem to be a natural choice to overcome efficiently the problem of artificial, discretization induced oscillations in cohesive fracture propagation. However, the moving crack front requests a continuously adapted change of the discretization throughout the simulation to ensure a sufficient locally refined cohesive process zone. Following the conventional *hp*-idea, where coarse elements are replaced during runtime by a finer mesh, is challenging and demands for a very sophisticated discretization kernel [42, 46, 47].

A simpler alternative is the multi-level *hp*-version of the finite element method recently introduced in [48]. Instead of the classical refine-by-replacement approach, the proposed *hp*-formulation refines a coarse base discretization by superposing a finer overlay mesh in the domain of interest. This refine-by-superposition idea dates back to the pioneering work of Mote [49] and has been applied successfully in the context of e.g. the hierarchic finite element method [50, 51], embedded localization zones [52–54], the spectral overlay method [55], the *hp-d*-refinement method [56], the *s*-version of the finite

element method [57–60], and the adaptive local overlapping grid method [61]. Partition of Unity methods (PUM, [62]), like the eXtended- and the generalized FEM [63, 64], also apply a similar concept by introducing additional shape functions to capture special solution characteristics.

The multi-level  $hp$ -method utilizes the refine-by-superposition concept to formulate an  $hp$ -version of the finite element method, which circumvents the implementational difficulties caused by the replacement of elements and which yields the same approximation accuracy as conventional  $hp$ -methods. In particular, the refine-by-superposition paradigm allows for a dynamic change of the discretization during the simulation by simply moving the position of the overlay mesh while keeping the base mesh unchanged.

Within this manuscript we demonstrate that the multi-level  $hp$ -method is ideally suited for adaptively resolving the cohesive zone ahead of the propagating crack tip with a high accuracy and low computational cost. In particular, the dynamic mesh capabilities of the approach allow for a local refinement of the discretization that follows the propagating crack tip. In this way, the domain of refinement can be decoupled from the total length of the potential crack path without degrading the approximation quality.

The present work is organized as follows: after a brief recapitulation of the cohesive zone model in Section 2, the essential ideas of the multi-level  $hp$ -approach are outlined in Section 3, and the use of the refinement method for the simulation of cohesive crack propagation is described. In Section 4, we systematically study the influence of  $h$ -,  $p$ -, and  $hp$ -refinement on the approximation quality in the context of different two- and three-dimensional examples. Further, we shown that the refinement method can be combined beneficially with the finite cell method (FCM) to simulate delamination within geometrically complex domains without the need for an explicit mesh representation. Finally, we summarize the main findings and conclude in Section 5.

## 2 Numerical simulation of cohesive crack growth

Within the cohesive zone model, the crack is modeled as a sharp interface immersed in the physical domain  $\Omega$ . We assume that the path  $\Gamma_C$  along which the crack potentially evolves is known in

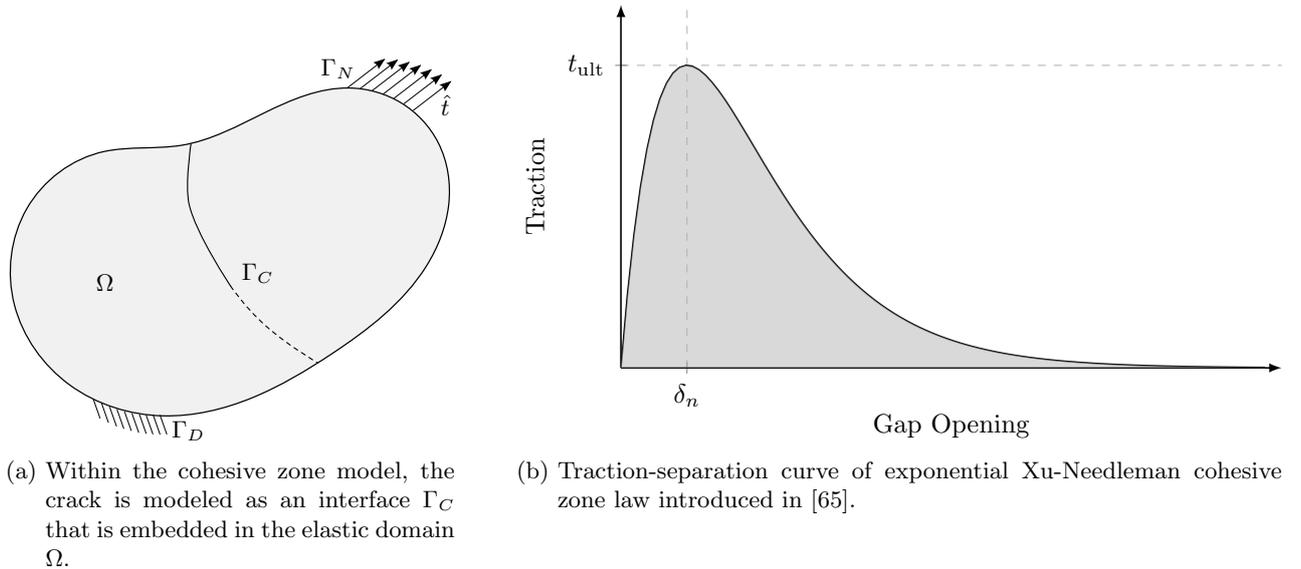


Figure 1: Cohesive crack growth

advance *cf.* Figure 1a. This is reasonable when considering e.g. the delamination of composite structures or cracks along material interfaces. Across the cracked interface, the displacement  $\mathbf{u}$  may be discontinuous yielding a crack opening  $\llbracket \mathbf{u} \rrbracket$ . The resulting gap can be split into a normal and tangential part  $\llbracket u_n \rrbracket$  and  $\llbracket \mathbf{u}_s \rrbracket$ , respectively. Pure normal separation is referred to as mode I fracture, whereas the shear separations represent sliding fracture (mode II) and tearing fracture (mode III). In the following, we discuss the equations governing the cohesive crack growth and the use of the finite element method for the numerical simulation of this process.

## 2.1 Governing equations

Assuming linear elasticity, the deformation of the continuum is described by the following set of partial differential equations:

$$\nabla \cdot \boldsymbol{\sigma} + \hat{\mathbf{b}} = \mathbf{0} \quad \text{on } \Omega \quad (1a)$$

$$\boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\varepsilon} \quad \text{on } \Omega \quad (1b)$$

$$\boldsymbol{\varepsilon} = \frac{1}{2} \left( \nabla \mathbf{u} + \nabla \mathbf{u}^\top \right) \quad \text{on } \Omega \quad (1c)$$

$$\mathbf{u} = \hat{\mathbf{u}} \quad \text{on } \Gamma_D \quad (1d)$$

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \hat{\mathbf{t}} \quad \text{on } \Gamma_N \quad (1e)$$

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{t}(\llbracket \mathbf{u} \rrbracket) \quad \text{on } \Gamma_C. \quad (1f)$$

Here,  $\boldsymbol{\sigma}$  describes the stress tensor,  $\hat{\mathbf{b}}$  the prescribed body force,  $\mathbf{C}$  the elasticity tensor,  $\boldsymbol{\varepsilon}$  the linear strain tensor,  $\mathbf{u}$  the displacement vector,  $\mathbf{n}$  the surface normal vector,  $\hat{\mathbf{u}}$  the prescribed displacement on the Dirichlet boundary  $\Gamma_D$ , and  $\hat{\mathbf{t}}$  the prescribed surface traction on the Neumann boundary  $\Gamma_N$ .

The traction-separation law  $\mathbf{t}(\llbracket \mathbf{u} \rrbracket)$  in (1f) extends the conventional elasticity problem to account for a cohesive crack analysis. This second constitutive equation models the load-carrying capabilities of the partially opened crack with respect to the crack opening  $\llbracket \mathbf{u} \rrbracket$ . A number of different traction-separation laws are discussed in the literature that are tuned to the respective application under consideration. A comprehensive review is given in e.g. [21, 66].

This work considers the so-called exponential cohesive zone law introduced in [65] and later improved in [66]. Like in most models, the interface traction is split into its normal and shear components  $t_n$ ,  $t_{s_1}$  and  $t_{s_2}$ , respectively. The studies presented in [22, 67] show that the cohesive zone of a mode I fracture typically is significantly smaller than for a mode II or a mode III failure. Therefore, the analysis presented in this work focuses on pure mode I failure since it is the computationally more challenging fracture type and thus serves as a severe test case for the proposed refinement scheme. Accordingly, we assume the tangential gap opening and the tangential traction in the interface to be zero. Following e.g. [66], the interface traction can be expressed as

$$t_n = \frac{G_c}{\delta_n} \frac{\llbracket u_n \rrbracket}{\delta_n} \exp \left( -\frac{\llbracket u_n \rrbracket}{\delta_n} \right) \quad \text{with } \delta_n = \frac{G_c}{t_{\text{ult}} \exp(1)}. \quad (2)$$

The depiction of this traction-separation law in Figure 1b reveals a gradual increase of the traction between the surfaces of the crack interface until the limit strength  $t_{\text{ult}}$  of the material is reached. Beyond this point, the fracture strength gradually decays to zero modeling the cohesive behavior of the fracture. The area enclosed by the traction-separation curve represents the energy release per unit of cracked surface and is denoted as fracture toughness  $G_c$ .

## 2.2 Finite element discretization

The numerical simulation of the crack propagation is based on the weak form of the governing partial differential equation (1):

Find  $\mathbf{u} \in \mathcal{V}$ , such that

$$\int_{\Omega} \delta \boldsymbol{\varepsilon} : \boldsymbol{\sigma} \, d\Omega + \int_{\Gamma_C} \delta [[\mathbf{u}]] \cdot \mathbf{t}([[ \mathbf{u} ]]) \, d\Gamma = \int_{\Omega} \delta \mathbf{u} \cdot \hat{\mathbf{b}} \, d\Omega + \int_{\Gamma_N} \delta \mathbf{u} \cdot \hat{\mathbf{t}} \, d\Gamma \quad \forall \delta \mathbf{u} \in \mathcal{V}_0, \quad (3)$$

with the functions spaces being defined as

$$\mathcal{V} = \{ \mathbf{v} : v_i \in H^1(\Omega), v_i|_{\Gamma_D} = \hat{u}_i \} \quad \text{and} \quad \mathcal{V}_0 = \{ \mathbf{v} : v_i \in H^1(\Omega), v_i|_{\Gamma_D} = 0 \}, \quad (4)$$

and  $H^1$  denoting the Sobolev space of degree one.

To solve the above problem numerically, we use the finite element method [68, 69]. Thereby, we assume that the crack interface  $\Gamma_C$  separates the original domain  $\Omega$  into two parts. Each of these sub-domains is discretized using a separate finite element mesh. For simplicity, we ensure that these meshes are matching geometrically along  $\Gamma_C$ . Using this discretization, the displacement field  $\mathbf{u}$  of the two sub-domains is given by the product of the shape function matrix  $\mathbf{N}$  and the degree of freedom vector  $\hat{\mathbf{u}}$ :

$$\mathbf{u}_i = \mathbf{N}_i \cdot \hat{\mathbf{u}}_i, \quad (5)$$

where the sub-index  $i$  of the above terms label the two different domains. In analogy, the strain resulting from the displacement is given by:

$$\boldsymbol{\varepsilon}_i = \mathbf{B}_i \cdot \hat{\mathbf{u}}_i, \quad (6)$$

where  $\mathbf{B}$  denotes the strain-displacement matrix [69]. The current gap value  $[[\mathbf{u}]]$  is computed as the difference between the displacements of the two sub-domains:

$$[[\mathbf{u}]] = \mathbf{u}_1 - \mathbf{u}_2 = \mathbf{N}_1 \cdot \hat{\mathbf{u}}_1 - \mathbf{N}_2 \cdot \hat{\mathbf{u}}_2 = [\mathbf{N}_1 \quad -\mathbf{N}_2] \cdot \begin{bmatrix} \hat{\mathbf{u}}_1 \\ \hat{\mathbf{u}}_2 \end{bmatrix} = \mathbf{H} \cdot \hat{\mathbf{u}}, \quad (7)$$

$$\text{with } \mathbf{H} = [\mathbf{N}_1 \quad -\mathbf{N}_2] \quad \text{and} \quad \hat{\mathbf{u}} = \begin{bmatrix} \hat{\mathbf{u}}_1 \\ \hat{\mathbf{u}}_2 \end{bmatrix}. \quad (8)$$

With this kinematic relations, we can translate the weak form (3) into the following system of non-linear equations

$$\mathbf{f}_{\text{int}}(\hat{\mathbf{u}}) = \mathbf{f}_{\text{ext}}, \quad \text{with} \quad \mathbf{f}_{\text{int}}(\hat{\mathbf{u}}) = \int_{\Omega} \mathbf{B}^\top \boldsymbol{\sigma} \, d\Omega + \int_{\Gamma_C} \mathbf{H}^\top \mathbf{t}([[ \mathbf{u} ]]) \, d\Gamma \quad (9)$$

$$\text{and} \quad \mathbf{f}_{\text{ext}} = \int_{\Omega} \mathbf{N}^\top \hat{\mathbf{b}} \, d\Omega + \int_{\Gamma_N} \mathbf{N}^\top \hat{\mathbf{t}} \, d\Gamma. \quad (10)$$

We use a Newton-Raphson iteration scheme to solve the non-linear equations. A common approach to account for the contributions of the progressing crack is to introduce additional, zero-thickness cohesive interface elements along  $\Gamma_C$ , whose kinematic is given by  $\mathbf{H}$ , *cf.* [70]. In this work, we follow a different approach. Instead of defining a new element type with kinematics specific to the gap opening, we compute the virtual work of the crack opening by numerically integrating the interface contribution along  $\Gamma_C$ . To this end, the crack interface is partitioned into a set of

integration domains that conform to the boundaries of the finite elements along  $\Gamma_C$ . On each of these integration segments, we distribute an appropriate number of Gauss-Legendre quadrature points to compute numerically the contributions of the progressing crack to the tangential stiffness matrix and the residual vector. This approach originates from the idea of weak mesh coupling [71, 72]. It is also similar to the Gauss-point-to-surface contact algorithm, which was used recently in [28] to simulate interface debonding. Mathematically, the used approach is equivalent to the conventional idea of interface elements. However, it has the advantage that it separates the computation of the interface contributions from the actual discretization. This simplifies the refinement process as no kinematics of refined interface elements has to be formulated.

### 2.3 Discretization induced oscillations

The solution procedure of the non-linear system of equations is numerically challenging since the cohesive interface approach suffers from an intrinsic discretization sensitivity as reported in e.g. [16–26, 28, 29]. This sensitivity is characterized by unphysical oscillations in the global load-displacement curve, which result from a cyclic stiffening phenomenon as sketched in the following, *cf.* Figure 2:

- we assume in load step  $k$  that the crack arrived at a node  $n_i$  of an undamaged element  $e_j$ , *cf.* Figure 2b. In the following load step increment  $k + 1$ , the node  $n_i$  will be displaced such that the crack propagates through  $e_j$ .
- using a linear displacement approximation, the crack spreads over the complete element domain. Thus, in large parts of element  $e_j$ , the numerically derived gap opening  $\delta_h$  is significantly larger than the analytical opening  $\delta_{ex}$  as depicted in Figure 2b.
- due to the higher stiffness of the interface material in the pre-cracked regime, *cf.* Figure 2c, the opening over the complete finite element requires a higher force than a local propagation of the crack, i.e.  $t_h \gg t_{ex}$ . This renders the discrete interface stiffer than in reality. Therefore, the element remains closed until the load is further increased, followed by an abrupt opening and a corresponding energy release. Thus the crack advances in discrete steps governed by the element size.

In between these steps, the high force required to break the interface causes a correspondingly large deformation of the elastic structure. This unrealistic deflection decreases again once the interface element has failed. The unphysical deformation cycles yield oscillations in the global load-displacement curve and may cause a series of snap-through and snap-back effects. These effects render the numerical solution of the non-linear system significantly more difficult and typically demand for arc-length based path-following techniques.

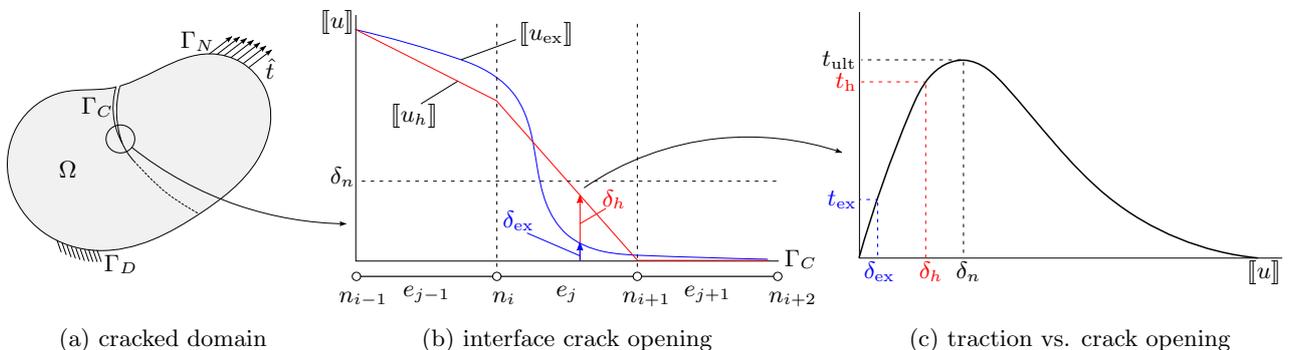


Figure 2: Discrete cohesive crack opening: linear approximation error.

The artificial oscillations become more pronounced with increasing element size. Thus, the accurate simulation of the crack propagation demands for a fine discretization of the cohesive zone. To this end, the following section outlines the essential idea of the multi-level  $hp$ -method, which allows for a non-uniform mesh refinement that follows the propagating crack tip.

### 3 Multi-level $hp$ -refinement

The prevalent idea when implementing  $hp$ -finite elements is to refine the discretization in the domain of interest by *replacing* the original, coarse elements with a set of finer elements [39, 73–77]. Although this approach has proven to perform well, its implementation is highly complex [46]. For this reason, most  $hp$ -discretization kernels are limited to *a priori* refinement which does not change during runtime. However, such a *static* discretization is not suited for the simulation of a progressing crack growth as the refinement zone cannot follow the propagating crack tip. Instead, a *dynamic* discretization is required, in which the mesh can be adapted in each analysis step.

The recently developed multi-level  $hp$ -method has proven this flexibility for a number of cases in static and dynamic analyses. The method was introduced in [48] and extended to three-dimensional meshes in [78]. In the following, we briefly outline the basic idea of this approach.

#### 3.1 Basic concept

An alternative to the classical refine-by-replacement approach is to increase the accuracy by *superposing* a coarse base mesh with a finer overlay mesh in the domain of interest *cf.* Figure 3. This hierarchical split decomposes the approximation  $u$  in a base part  $u_b$  and an overlay part  $u_o$ :

$$u = u_b + u_o. \quad (11)$$

Thereby, the essential idea is that the overlay solution  $u_o$  is zero on the boundary of the refinement zone. In this way, the  $C^0$ -continuity of the final approximation  $u$  is ensured by construction. This idea is illustrated in Figure 3.

The multi-level  $hp$ -method uses this paradigm to formulate an easy to implement and highly flexible  $hp$ -discretization that yields the same high approximation quality as conventional  $hp$ -approaches [48, 78]. To this end, the domain is first discretized by a base-mesh consisting of coarse  $p$ -elements [79, 80]. In contrast to the classically used Lagrange polynomials, these elements employ integrated Legendre shape functions for the high-order approximation of the solution. These shape functions are designed to minimize the condition number of the stiffness matrix. Additionally, this type of shape functions has two further advantages that are important in the context of the multi-level  $hp$ -refinement.

First, the set of shape functions is hierarchical in the sense that the approximation order is increased by simply adding new higher-order modes to an existing low order basis. This idea is illustrated in

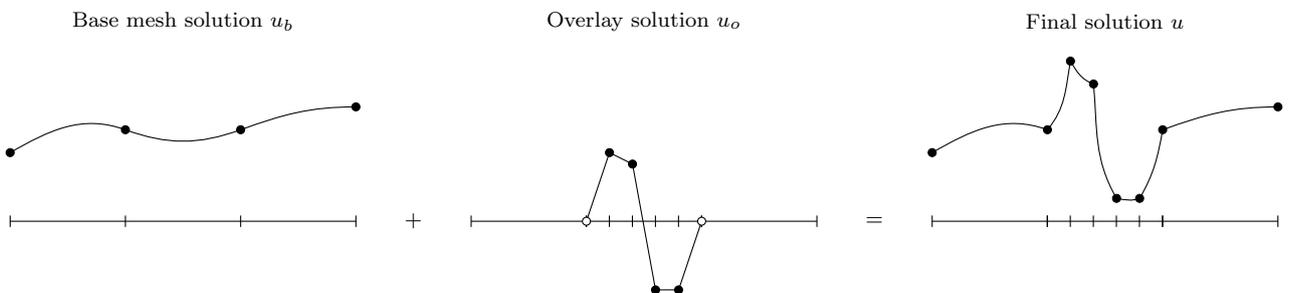


Figure 3: Conceptual idea of the refinement by superposition following [56].

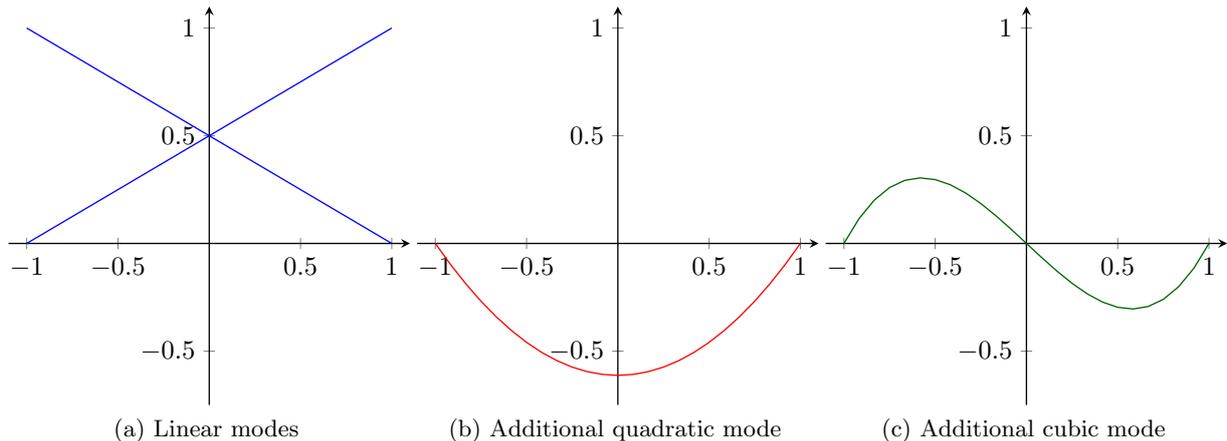


Figure 4: Hierarchical integrated Legendre basis functions [79, 80].

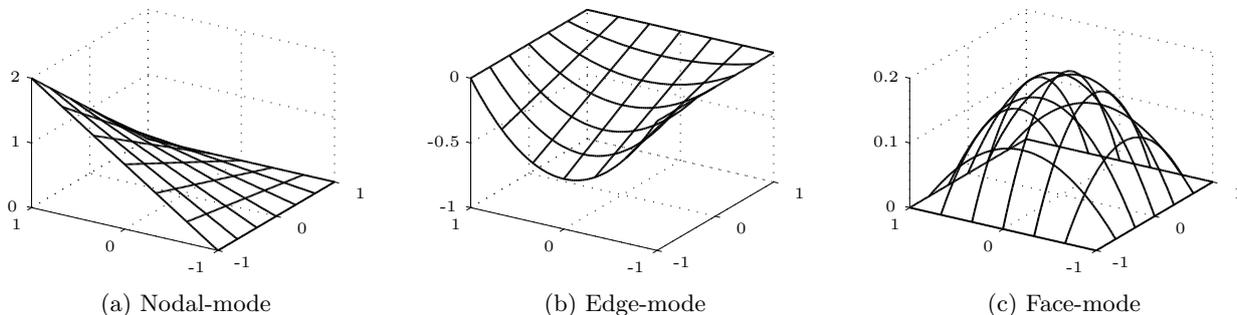


Figure 5: Two-dimensional mode types following e.g. [79, 80].

Figure 4. As long as this basis contains the two linear shape functions, it is complete even if high-order shape functions are removed. This is fundamentally different in the classical Lagrange basis, in which a complete new set of basis functions is used when increasing the approximation order. Hence, removing selected shape functions from the set of basis function would destroy the completeness.

Second, the integrated Legendre basis functions can be associated directly with the nodes, edges, faces and solids of the mesh in the sense that the functions are non-zero on only one of these components and zero on all others as illustrated in Figure 5. This mode-concept allows to easily remove selected shape functions from the approximation space by “deactivating” the associated topological component.

The essential idea of the multi-level  $hp$ -approach is to refine this coarse base-discretization by superposing a set of finer overlay elements in the domain of interest as illustrated in Figure 6. The refinement can be repeated hierarchically by recursively superposing multiple levels of finer elements giving the method its name. This recursive structure gives rise to a refinement tree, in which the base element is the root and the sub-elements are the children. The last level of sub-elements is denoted as leaf-elements.

When following this hierarchical refinement approach, it is essential to ensure both, linear independence and compatibility of the basis functions. Due to the clear tree-structure of the refinement and the nature of the used shape functions, these requirements can be met without difficulty:

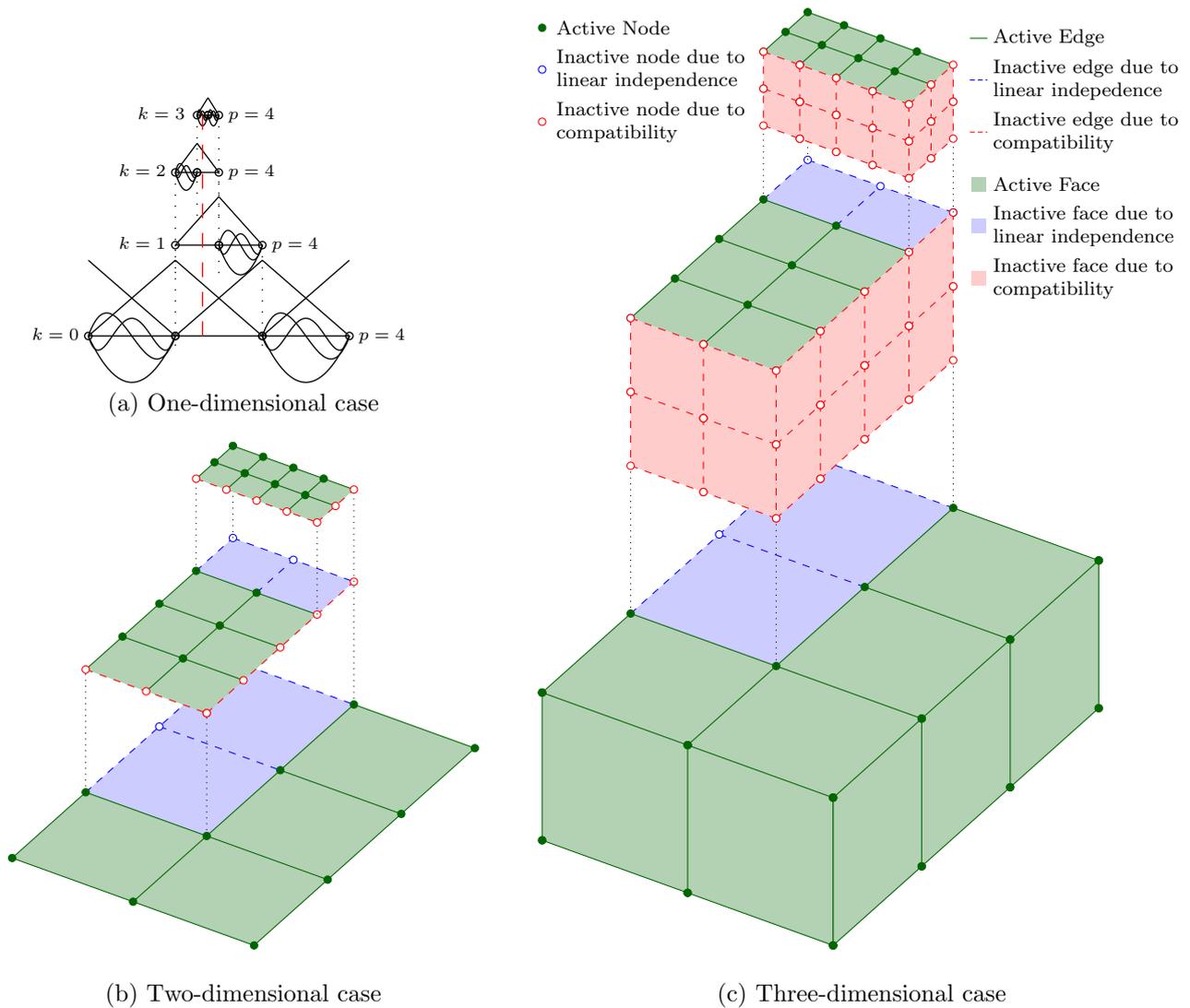
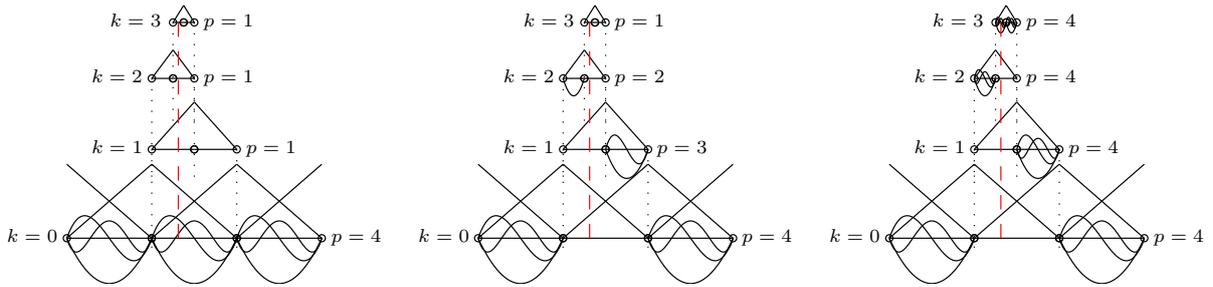


Figure 6: Conceptual idea of the multi-level  $hp$ -method following [48].

**Linear independence** To provide a linear independent basis, it must be ensured that no shape function can be represented by a sub-set of the applied shape functions. Due to the hierarchical nature of the integrated Legendre shape functions, this can be done easily by ensuring that each high-order mode is only present once in each branch of the refinement tree. For the example depicted in Figure 6, this is done by placing all high-order modes on the leaf-elements of the tree. On the respective parent-levels, only nodal modes remain active to ensure the global continuity and the completeness of the approximation. Technically, this is implemented by deactivating edges, faces, and solids that have been refined.

**Compatibility** For the compatibility of the discretization it has to be ensured that all shape functions are  $C^0$ -continuous between adjacent elements [68, 81]. This is of significant difficulty in the classical refine-by-replacement approach as the discretization is non-matching at the transition between the refined and the coarse parts of the mesh. However, in the used refine-by-superposition strategy, the continuity between the coarse and refined zones can be ensured easily by simply applying homogeneous Dirichlet boundary conditions on the overlay solution  $u_o$ . In this way, the overlay



(a) *hp-d*-approach [56, 82]: linear over-layer functions on a high-order base mesh. (b) Multi-level *hp*-approach with a linear  $p$ -distribution. (c) Multi-level *hp*-approach with a uniform  $p$ -distribution.

Figure 7: Comparison of different approaches for hierarchical, high-order refinement.

solution is zero on the boundary of the refinement zone, which ensures the inter-element continuity as illustrated in Figure 3. To enforce these conditions, all overlay shape functions that are non-zero on the boundary of the refinement zone have to be removed from the approximation space. Due to the aforementioned mode concept, this can be done by simply deactivating all topological components on the boundary of the overlay mesh. The problem of hanging nodes is avoided by construction, as no degrees of freedom are present on the boundary of the refinement zone. This is depicted in Figure 6.

In addition to the element size  $h$ , the overlay approach also allows to adapt the polynomial order of the element shape functions over the different mesh levels. One possibility depicted in Figure 7a is to use only linear shape functions on the overlay elements. Here, the high-order shape functions can remain active on the base mesh. In this configuration, the multi-level *hp*-approach turns into the *hp-d* method, which was introduced in [56] and extended in [82–89]. The aforementioned alternative is to deactivate the high-order modes on refined elements and instead to activate them on the leaf-elements of the refinement tree. Thereby, the approximation order can be distributed uniformly by setting the same value of  $p$  on all leaf-elements. A corresponding setup is depicted in Figure 7c. Alternatively, the polynomial degree might also be distributed linearly over the refinement levels as illustrated in Figure 7b. This setup follows the classical idea of a *hp*-mesh design, in which the element size is graded geometrically towards a singularity and the approximation order is increased linearly away from the singularity [38]. This comparison demonstrates that the refine-by-superposition concept allows for the same flexibility in the choice of  $h$  and  $p$  as the conventional refine-by-replacement approach.

### 3.2 Implementation aspects

In the previous section, we showed that the main challenge of the refinement-by-superposition approach is to ensure linear independence and compatibility of the basis functions. To this end, the “correct” nodes, edges, and faces of the mesh have to be deactivated. The following section outlines a possible implementation of this deactivation process in the context of an object-oriented finite element framework. For this purpose, we first introduce the used data structure in Section 3.2.1. In Section 3.2.2, we describe the refinement procedure and the necessary data handling for deactivating the “correct” nodes, edges, and faces. To limit the complexity of the illustration, the respective depictions focus on the refinement of two-dimensional, quadrilateral elements. However, this is no limitation of the algorithm itself, and an extension to three-dimensional elements follows in analogy.

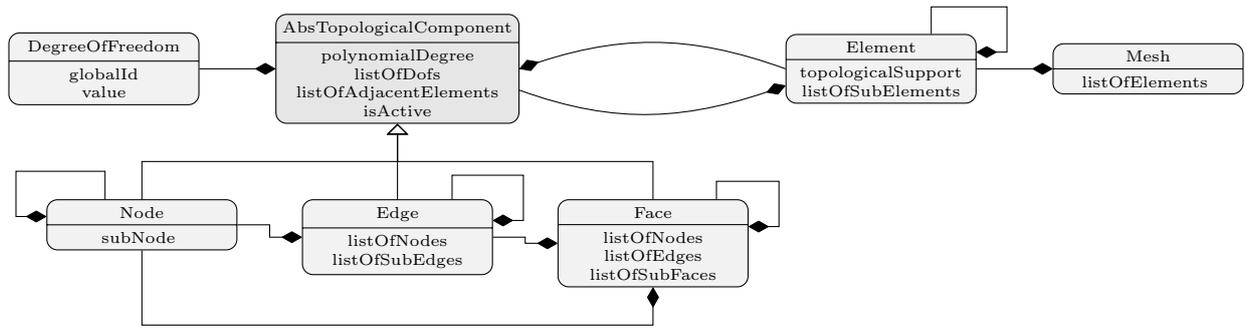


Figure 8: UML representation of the multi-level  $hp$ -data structure following [48].

### 3.2.1 Data structure

Within this work, we assume that the connectivity of the finite element mesh is formulated in terms of three types of topological components: the nodes, the edges, and the faces. In the context of an object oriented framework, each of these component types is represented by one corresponding class. As depicted in Figure 8, an edge consists of a list of nodes (typically two), and a faces consists of a list of nodes and a list of edges. To represent the hierarchical refinement tree mentioned in the previous section, each topological component additionally stores a reference to its sub-components that are created in the refinement process explained in Section 3.2.2. In the case of a node, this is one sub-node, whereas edges and faces store a list of sub-edges and sub-faces, respectively.

To treat the topological components in a common way, the three classes inherit from a common abstract base class `AbsTopologicalComponent`. In this way, large parts of the implementation can be re-used by placing it in the common base class. In particular, this applies to the degree of freedom handling. As outlined in the previous section, the shape functions and corresponding degrees of freedom are identified as nodal, edge, or face modes. For this reason, each topological component stores a polynomial degree that specifies the order of the associated shape functions. Furthermore, the topological components are responsible for handling the degrees of freedom. To this end, the components store a list of degrees of freedom. Given that an `isActive` flag is set, this list is filled during the setup phase of the program taking into account the associated polynomial degree.

Using the described topological setup, a finite element can be defined by specifying one topological component as its *support*. In the one-dimensional case, the elements are defined on edges, whereas in two dimensions, a face serves as the support. As depicted in Figure 8, each element stores a reference to the respective topological support, and vice versa each topological component holds a list of all adjacent elements. Using the topological support, the element can build up the location map and construct the element shape functions depending on the polynomial degree specified by the topological support. Additionally, each element holds a list of sub-elements, which overlay the respective element in the next refinement level. This allows the refinement tree to be build up recursively starting from the coarse base elements.

The list of all base elements is stored in the mesh class. The main responsibility of the mesh class is to evaluate the element shape functions at the quadrature points. In this way, the element stiffness matrices can be integrated and then assembled into the final system matrix.

### 3.2.2 Refinement procedure

Using the aforementioned data structure, the discretization can be refined by performing the four steps depicted in Figure 9, which are explained in the following.

The first step is to mark all elements that shall be refined or coarsened. This can be done based

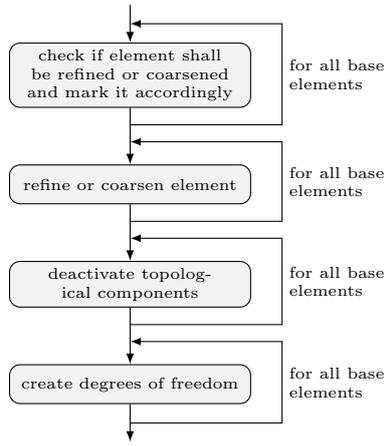


Figure 9: Schematical representation of the multi-level  $hp$ -refinement procedure.

on an *a posteriori* error estimator (see e.g. [74, 90]) or based on geometric information such as the position of re-entrant corners or singular loads. A description of how we select the elements for refinement in the simulation of delamination processes is given in Section 3.3.

The second step of the refinement procedure is the actual refinement of the elements. To this end, each base element is passed to the `refineElement` function outlined in Algorithm 1. This function first checks, whether the passed element is marked for refinement. If this is the case, the new sub-elements are created by calling the function `createSubElements` on the passed element *cf.* Algorithm 2. Afterwards, the `refineElement` function is called recursively on each of the created sub-elements. In this way, the function-call traverses naturally through the complete refinement tree in a depth-first pattern.

The process of creating the sub-elements is illustrated in Algorithm 2. Here, the first step is to create the new sub-topology. To this end, the topological support of the element is passed to the

---

**Algorithm 1** Recursive refinement of an element

---

```

1 function refineElement( elementToBeRefined )
2 {
3     // check whether the passed element shall be refined
4     if( isElementSelectedForRefinement( elementToBeRefined ) == true )
5     {
6         // create new sub-elements -> see Algorithm 2
7         createSubElements( elementToBeRefined )
8
9         // traverse through the refinement-tree by recursively
10        // calling the refinement function on each sub-element
11        for each subElement in elementToBeRefined->listOfSubElements
12        {
13            refineElement( subElement )
14        }
15    }
16    // check whether the passed element shall be coarsened
17    else if( isElementSelectedForCoarsening( elementToBeRefined ) == true )
18    {
19        // remove the sub-elements from the current element -> see Algorithm 3
20        removeSubElements( elementToBeRefined )
21    }
22 }
  
```

---

---

**Algorithm 2** Creation of sub-elements
 

---

```

1 function createSubElements( elementToBeRefined )
2 {
3   // first, refine the topological support of the element ...
4   topologicalSupport = elementToBeRefined->topologicalSupport
5   createSubTopology( topologicalSupport )

7   // ... and then create new elements on each of the sub-topologies
8   for each subTopology in topologicalSupport->getListOfSubTopologies
9   {
10    // first, create the new sub-element on the sub-topology ...
11    subElement = createElement( subTopology )

13    // ... and add new element to the list of sub elements of the parent
14    elementToBeRefined->addSubElement( subElement )

16    // then de-register the parent element from the adjacency
17    // list of the sub-topology ...
18    subTopology->deregisterAdjacentElement( elementToBeRefined )

20    // ... and register the sub-element instead
21    subTopology->registerAdjacentElement( subElement )
22  }
23 }

```

---

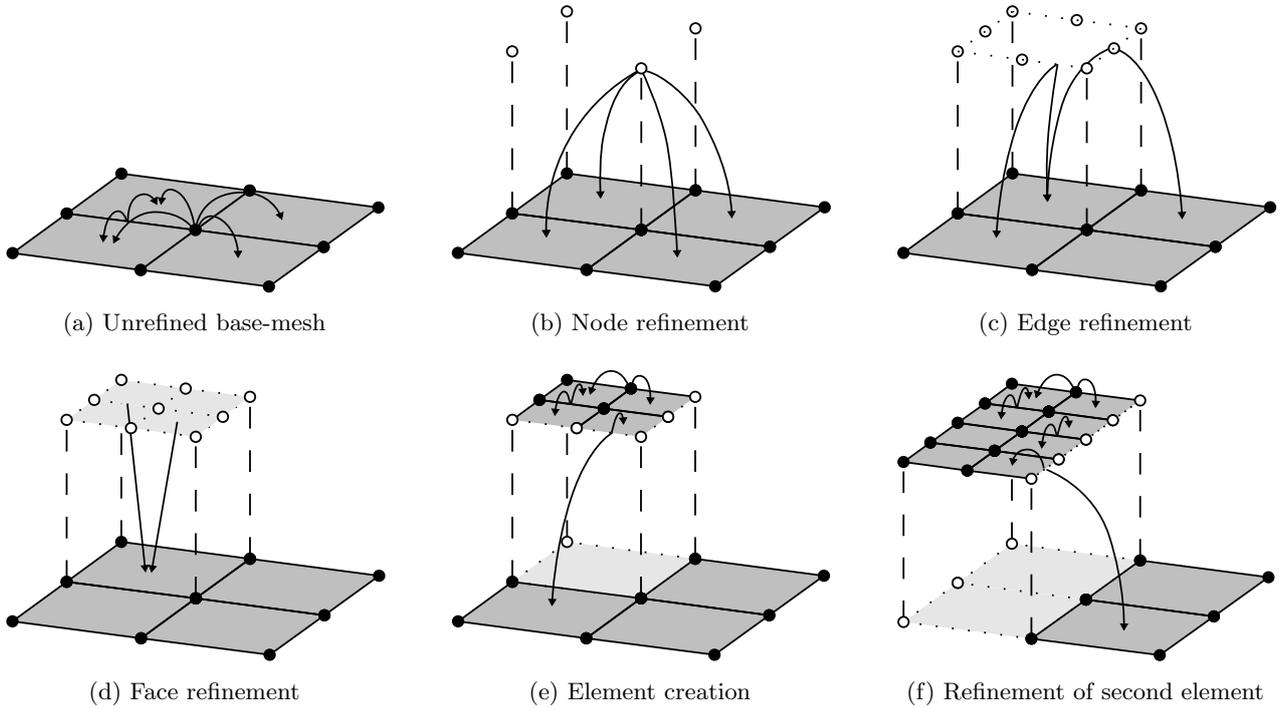


Figure 10: Creation of overlaying sub-topologies and sub-elements following [48]. The arrows depict the adjacency relation between the created topological sub-components and the elements.

function `createSubTopology`. The creation of the overlaying sub-topology is depicted in Figure 10. In the figures, the arrows exemplarily illustrate the list of adjacent elements for selected topological components.

As depicted in Figure 10b, the first step of refining the topological support of the element (in the present example a quadrilateral face) is to refine the nodes of the topological component. To this end, a new sub-node is added to each node of the topological support. These sub-nodes are geometrically identical to their parents and thus overlay them in the next refinement level. Furthermore, the sub-nodes inherit the list of adjacent elements from their parents. Therefore, at this point the sub-nodes are still considered to be adjacent to the original coarse elements. In Figure 10b, this is illustrated for the central sub-node.

In analogy to the nodes, the edges and the face itself are refined by creating new topological sub-components that overlay their parents in the next refinement level (see Figures 10c and d). Again, these new topological components inherit the adjacency list from their parents and therefore are still regarded as being connected to the coarse base elements.

After the sub-topology has been created, new sub-elements can be created. To this end, Algorithm 2 loops over all the sub-components of the topological support. In the present case of a quadrilateral face, these sub-components are the four sub-faces. In the case of a hexahedral topological support, the sub-components would be eight octants. On each of these sub-components, a new element is created (Algorithm 2, line 11) and added as a sub-element to the parent element. It is important to remember that these sub-elements do not replace their parents but overlay them in the next refinement level. Therefore, the coarse base elements are still part of the discretization. However, the new sub-elements change the adjacency relation of the topological components. As indicated in the final part of Algorithm 2, the parent elements are de-registered from the adjacency list of the sub-topology and the new sub-elements are registered instead. Thereby, the functions `deregisterAdjacentElement` and `registerAdjacentElement` do not only affect the four sub-faces but also the connected nodes and edges. The updated adjacency relation are depicted Figure 10e.

If the next element of the base mesh is refined, the described procedure is repeated. Thereby, common topological components that have been refined before are not refined a second time. Instead, the previously created sub-components are used again. In this way, a topologically connected overlay mesh emerges. Also the adjacency relations are updated accordingly, as illustrated in Figure 10f.

In analogy to the refinement, the discretization can also be coarsened easily as illustrated in Algorithm 3. To this end, the depth of the refinement tree is decreased by recursively deleting the leaf-elements and updating the adjacency list of the topology.

As depicted in Figure 9, the third step of the refinement procedure is the deactivation of certain topological components to ensure the compatibility and the linear independence of the basis functions. As outlined in the previous section, the compatibility requires to de-activate all topological components on the boundary of the refinement zone. Using the aforementioned data setup, these components can be identified easily by comparing the refinement level of all adjacent elements. This is done in the first part of the function `deactivateTopology` outlined in Algorithm 4, which is called on the topological support of each base element of the mesh. In the second part of the algorithm, the function is called recursively on all the connected topological components. In the present case of a quadrilateral face, this concerns the four nodes and the four edges of the face. In the third step, the function `deactivateTopology` is again called recursively on all sub-topologies which were created during the refinement procedure. In this way, the function-call traverses naturally through the complete refinement tree and all topological components of the hierarchical mesh are reached. Finally, the passed topological component is deactivated if it is either on the boundary of the refinement zone or if all the sub-topologies are active. The first check ensure the compatibility of the basis functions, while the second check ensures the linear independence.

The final step of the refinement procedure depicted in Figure 9 is the creation of the degrees of freedom. As outlined in Section 3.2.1, the degrees of freedom are associated to the topological

---

**Algorithm 3** Coarsening of refinement tree

---

```
1 function removeSubElements( elementToBeCoarsend )
2 {
3     // first, coarsen all sub-elements of the sub-elements of the passed element
4     // To this end, traverse through the refinement-tree by recursively
5     // calling the coarsening function on each sub-element
6     for each subElement in elementToBeCoarsend->listOfSubElements
7     {
8         removeSubElements( subElement )
9     }
10
11    // then, de-register the direct sub-element of the passed element
12    // from the adjacency list of their topology
13    for each subElement in elementToBeCoarsend->listOfSubElements
14    {
15        deregisterElementFromTopologicalSupport( subElement )
16    }
17
18    // and finally delete the actual sub-elements
19    elementToBeCoarsend->listOfSubElements->clear()
20 }
```

components. Therefore, each topological component of the hierarchical mesh can allocate degrees of freedom depending of the associated polynomial degree, given that it has not been deactivated in Algorithm 4.

After completing this setup, the stiffness matrix and the load vector can be integrated. To this end, Gaussian quadrature points are distributed in each leaf-element of the refinement tree. On every quadrature point, all non-zero shape functions of all overlapping elements are evaluated and written into one shape function vector  $\mathbf{N}$ . In the same way, the derivatives of the shape functions are written into one strain-displacement matrix  $\mathbf{B}$ . With these matrices, the stiffness matrix and the load vector can be computed and integrated. After the finite element system of equations has been built and solved, the discretization can be changed again by performing the four refinement steps depicted in Figure 9 with an updated refinement criterion.

### 3.3 Multi-level $hp$ -refinement for cohesive crack growth

As outlined in the previous section, the refine-by-superposition concept applied by the multi-level  $hp$ -method allows for a dynamic change of the discretization during the simulation runtime. Within this work, we use this high flexibility to keep the refinement zone local to the crack tip, while it is propagating through the domain. To this end, we update the mesh in every load step of the non-linear solution scheme. Thereby, the normal gap opening  $[[u_n]]$  across the interface  $\Gamma_C$  serves as a natural refinement criterion. To mark the elements for refinement (step 1 in Figure 9), we distribute equidistant sample points in every element edge along  $\Gamma_C$ . The number of sampling points is chosen in dependence of the polynomial degree of the adjacent elements. On every sampling point we compute the normal gap value  $[[u_n]]$  according to (8). If the computed gap opening is within a range  $[g_1, g_2]$ , the leaf-elements adjacent to the current point are marked for refinement given that a pre-defined refinement depth is not exceeded. If the gap opening is larger than  $g_2$  for all sampling points of an element edge, the respective element is marked for being coarsened. In this way, the refinement tree builds up recursively and refines the mesh only around the crack tip.

The values of  $g_1$  and  $g_2$  are chosen empirically such that  $g_1 < \delta_n < g_2$ , where  $\delta_n$  is the gap opening of maximal cohesive traction *cf.* (2). The examples presented in the next section demonstrate that the refinement region resolves the fracture zone sharply when choosing  $g_1$  about two orders of

---

**Algorithm 4** De-activation of topological components to ensure the compatibility and the linear independence of the basis functions

---

```
1 function deactivateTopology( topologicalComponent )
2 {
3     // first, check if the passed topology is on the boundary of the refinement
4     // zone by running over each element adjacent to the component ...
5     topologicalComponentIsOnBoundary = false
6     for each adjacentElement in topologicalSupport->adjacentElements
7     {
8         // ... and comparing the refinement level of the adjacent element
9         // to the refinement level of the topological component
10        if( adjacentElement->refinementLevel !=
11            topologicalComponent->refinementLevel )
12        {
13            topologicalComponentIsOnBoundary = true
14            break
15        }
16    }
17
18    // then, do the same for all connected nodes and edges by
19    // calling deactivateTopology recursively on these components
20    for each connectedComponent in topologicalComponent->getConnectedComponents()
21    {
22        deactivateTopology( connectedComponent )
23    }
24
25    // and then do the same for all the sub-components recursively
26    areAllSubTopologiesActive = true
27    for each subTopology in topologicalSupport->getListOfSubTopologies()
28    {
29        deactivateTopology( subTopology )
30        areAllSubTopologiesActive = areAllSubTopologiesActive &&
31            subTopology->isActive
32    }
33
34    // and finally deactivate the passed component if it is either on the boundary
35    // of the refinement zone or if all the sub-components are active
36    if ( topologicalComponentIsOnBoundary == true ||
37        areAllSubTopologiesActive == true )
38    {
39        topologicalComponent->isActive = false
40    }
41    else
42    {
43        topologicalComponent->isActive = true
44    }
45 }
```

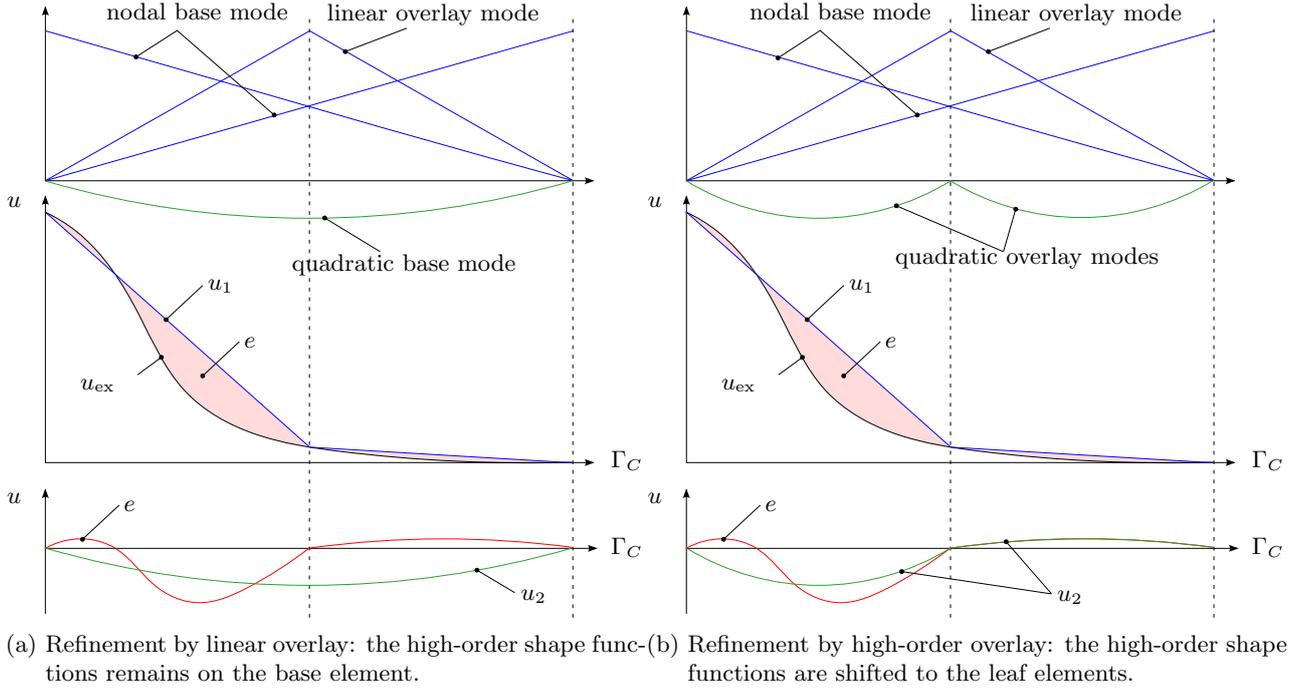


Figure 11: Approximation of solution by overlay-refinement with  $p = 2$ .

magnitude smaller and  $g_2$  about one order of magnitude larger than  $\delta_n$ . The specific values of  $g_1$  and  $g_2$  used for the different examples are given in the description of the respective example setup.

In addition to the refinement of the element size  $h$ , also the choice of the approximation order  $p$  is important for the result quality. As depicted in Figure 7, the overlay refinement approach allows to place the high-order modes on different levels of the refinement tree. The number of shape functions and with it the number of unknowns is the smallest when using a high-order base mesh and a linear overlay mesh  $cf.$  Figure 7a. However, the numerical studies presented in [48, 78] demonstrate that the approximation quality increases significantly when placing the high-order modes on the leaf elements  $cf.$  Figures 7b and c. For this reason, the first question to be analyzed in the examples section is which  $p$ -distribution yields the highest approximation quality for the simulation of cohesive crack propagation.

It is to be expected that the cohesive zone is resolved best when the high-order modes are placed on the leaf-elements. This can be followed from the exemplary scenario depicted in Figure 11a. Here, a standard quadratic base discretization is refined by adding an additional linear overlay mode. Hence, the final numerical solution is composed of a linear contribution  $u_1$  and a quadratic contribution  $u_2$ :

$$u_h = u_1 + u_2, \quad (12)$$

both having support on the full base element.

When the crack arrives at the considered element, these two contributions approximate the complex gap function  $[[u_{ex}]]$  that forms the transition from the closed to the fully opened crack. Using only the linear contribution for approximation results in an error

$$e = u_{ex} - u_1. \quad (13)$$

In the considered scenario, the residual is close to zero on the right half-interval but non-zero on the

left. The remaining deviation cannot be compensated by the quadratic mode  $u_2$  as this shape function spans the complete *coarse* element. Therefore, the left and the right part of the solution cannot be approximated independently. Instead, the solution scheme has to find the best-fit compromise.

As illustrated in Figure 11b, the situation is fundamentally different if the base-element is refined by placing the high-order modes on the leaf-elements. In this case, the discretization features two quadratic overlay modes, whose support is restricted to the left and right part of the base element, respectively. Therefore, these two modes can approximate the two parts of the solution at a much higher accuracy. For the present case, this allows to capture the right-hand-side residual with high accuracy, which confines the approximation error to the left element.

To verify this assumption, the influence of the element size  $h$ , the approximation order  $p$  and the overlay configuration on the result quality is systematically studied in the following examples.

## 4 Examples

In the following, we study the performance of the proposed multi-level  $hp$ -refinement method with several examples. Thereby, the potential of  $h$ -,  $p$ -, and  $hp$ -refinement for preventing discretization-induced oscillations is of particular interest. To this end, we first study the approximation quality of these refinement schemes on a pure elastic crack with a singular solution. Secondly, we analyze the approximation quality of the different discretizations using the well known double cantilever beam benchmark. In third example, the proposed refinement scheme is used in a three-dimensional simulation of a two-ply laminate. We close this section by analyzing the potential of the finite cell method to model cohesive crack propagation within complex domains. In all examples, we concentrate on pure mode I delamination as it is the numerically most challenging type of failure [22, 67].

### 4.1 Cracked panel benchmark

In this first example, we analyze the approximation quality of the suggested refinement schemes for crack problems. To this end, we use the cracked panel benchmark depicted in Figure 12a. The example is a severe test case with regard to the convergence behavior of the analysis since the pure

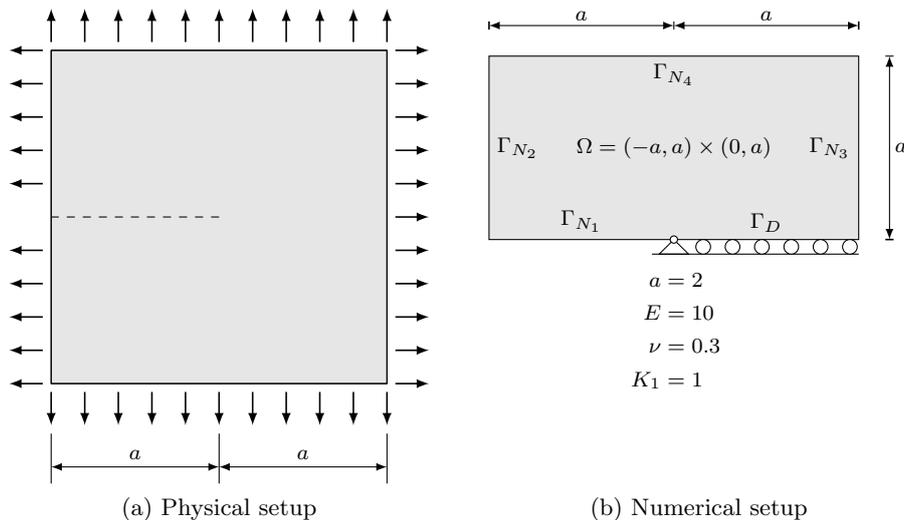


Figure 12: Setup of cracked panel (plane strain).

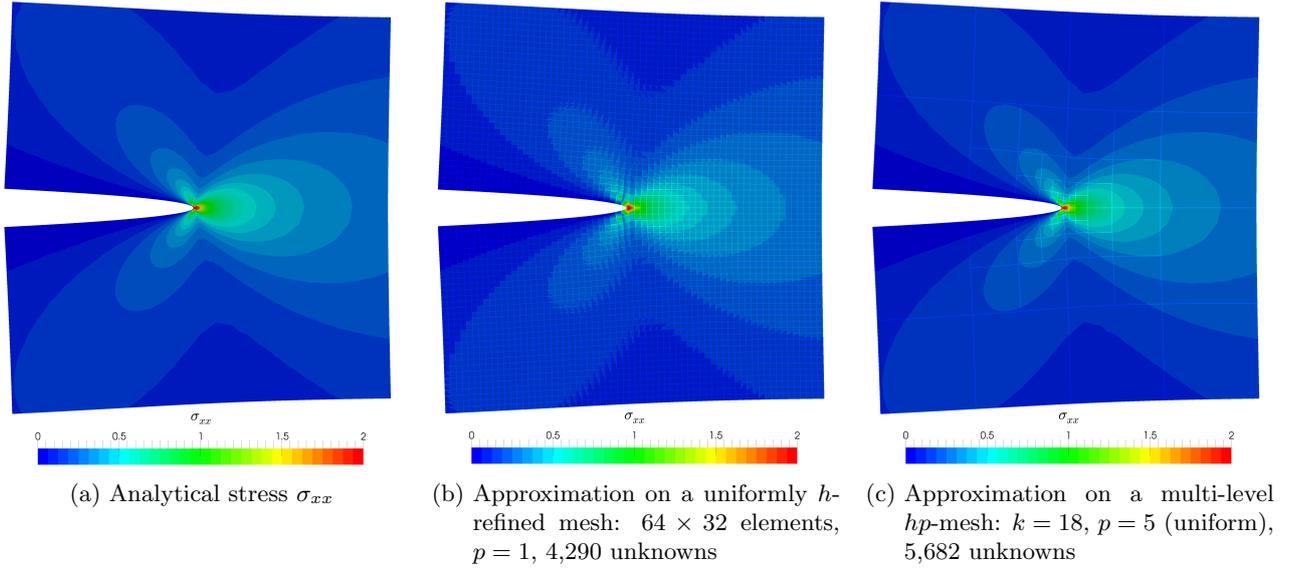


Figure 13: Analytical distribution of  $\sigma_{xx}$  and its approximation on different discretizations (due to symmetry, only the upper half is computed).

elastic stress distribution in the panel is analyzed instead of considering a cohesive crack growth. This renders the stress state singular at the crack tip, which is different in the case of a cohesive crack where the cohesive traction-separation law prevents the un-physical singular stress state [16, 26]. Thus, the present singular configuration is more challenging from a numerical point of view. It is assumed that the discretization leading to high quality results in the stress analysis is well suited also in a cohesive crack growth analysis.

Following e.g. [91], the analytical stress state of a pure mode I deformation under plane strain conditions is given by:

$$\sigma_{xx} = \frac{K_1}{\sqrt{2\pi r}} \cos \frac{\theta}{2} \left( 1 - \sin \frac{\theta}{2} \sin \frac{3\theta}{2} \right) \quad (14a)$$

$$\sigma_{yy} = \frac{K_1}{\sqrt{2\pi r}} \cos \frac{\theta}{2} \left( 1 + \sin \frac{\theta}{2} \sin \frac{3\theta}{2} \right) \quad (14b)$$

$$\sigma_{xy} = \frac{K_1}{\sqrt{2\pi r}} \sin \frac{\theta}{2} \cos \frac{\theta}{2} \cos \frac{3\theta}{2}. \quad (14c)$$

In the above equations,  $(r, \theta)$  are the polar coordinates, and  $K_1$  denotes the stress intensity factor, which is chosen as  $K_1 = 1.0$  the present example. The analytical stress distribution  $\sigma_{xx}$  is depicted in Figure 13a. Due to the leading  $1/\sqrt{r}$  term in (14), all three stress components tend to infinity as  $r$  approaches the origin. This renders the problem singular. Commonly, the strength of the singularity is characterized by an irregularity coefficient  $\lambda$  [80]. Following [91],  $\lambda$  assumes the value  $1/2$  for the present example.

#### 4.1.1 Numerical setup

For the approximation of the analytical solution with the finite element method, the symmetry of the problem is exploited by only considering the upper half of the panel. The respective symmetry Dirichlet boundary conditions are applied as depicted in Figure 12b. On the remaining boundary, the analytical stress distributions given in (14) are used as traction values of the Neumann boundary

conditions:

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{0} \quad \forall \mathbf{x} \in \Gamma_{N_1}, \quad \boldsymbol{\sigma} \cdot \mathbf{n} = - \begin{bmatrix} \sigma_{xx} \\ \sigma_{xy} \end{bmatrix} \quad \forall \mathbf{x} \in \Gamma_{N_2}, \quad (15a)$$

$$\boldsymbol{\sigma} \cdot \mathbf{n} = + \begin{bmatrix} \sigma_{xy} \\ \sigma_{yy} \end{bmatrix} \quad \forall \mathbf{x} \in \Gamma_{N_4}, \quad \boldsymbol{\sigma} \cdot \mathbf{n} = + \begin{bmatrix} \sigma_{xx} \\ \sigma_{xy} \end{bmatrix} \quad \forall \mathbf{x} \in \Gamma_{N_3}. \quad (15b)$$

The symmetric part of the domain is discretized with a initial mesh of  $4 \times 2$  elements. To account for the singular stress state at the crack tip, this base discretization is refined geometrically towards the crack tip using *hp-d*- and multi-level *hp*-overlay meshes, respectively. The resulting approximation to the analytical stress distribution  $\sigma_{xx}$  is exemplarily depicted in Figure 13c. The comparison to the approximation obtained by a uniform, low-order *h*-refinement depicted in Figure 13b demonstrates qualitatively the superiority of the *hp*-approach.

#### 4.1.2 Comparison of different refinement strategies

As outlined in Section 3.1, the multi-level *hp*-approach allows for different overlay strategies. In the following, we compare the approximation quality of the three refinement strategies depicted in Figure 7. Additionally, we considered a uniform *h*-refinement and a uniform order elevation.

For the comparison of the obtained result quality, the approximation error is analyzed in the energy norm

$$\|e\|_E = \sqrt{\frac{\Pi_{\text{ex}} - \Pi_{\text{fem}}}{\Pi_{\text{ex}}}} \cdot 100\%, \quad \text{with} \quad \Pi_{\text{ex}} \approx 0.23706469 \cdot K_1^2 \cdot \frac{a}{E} \quad (16)$$

Here,  $\Pi_{\text{ex}}$  and  $\Pi_{\text{fem}}$  denote the exact and the numerically computed strain energy, respectively. The reference value is taken from [91].

**Uniform *h*- and *p*-refinement** The simplest idea to increase the approximation quality is to uniformly decrease the element size *h* or to elevate the approximation order *p* of all elements. The error decrease resulting from these extensions are depicted in Figure 14a. The results show that a uniform *h*-refinement yields an algebraic convergence of the approximation error at a rate  $\beta = 0.25$ . This is in excellent agreement with the *a priori* estimates presented in e.g. [80, 92], which predict that the convergence rate in the presence of a singularity cannot be higher than  $\lambda/2$ . The convergence resulting from a uniform *p*-elevation is also algebraic but at a rate  $\beta = 0.49 \approx \lambda$ . These results also meet the *a priori* estimates given in e.g. [80, 92], which predict that a *p*-refinement achieves twice the convergence rate of a uniform *h*-refinement. This allows for a significantly higher accuracy using the same number of unknowns. However, both refinement approaches only yield an algebraic decay of the error. Therefore, the following three refinement strategies employ a non-uniform mesh-refinement to achieve an exponential convergence of the approximation error.

**Strategy 1: Uniform *p*-elevation with fixed number of linear overlay meshes** The first approach is the *hp-d*-refinement [56, 82]. Here, a high-order base mesh is used that is refined towards the crack tip by recursively superposing linear overlay meshes. An example of such an *hp-d*-discretization is depicted in Figure 15a. For the convergence curve presented in Figure 14a, we use an 18-times recursively refined mesh and elevate the approximation order *p* from one to ten. The results show a high convergence of the error for the first refinement steps. However, the rates of convergence gradually decay for the later steps, and the convergence turns algebraic in the asymptotic limit with  $\beta \approx \lambda$ . This demonstrates that refining the base discretization by linear overlay meshes does not improve the approximation quality substantially.

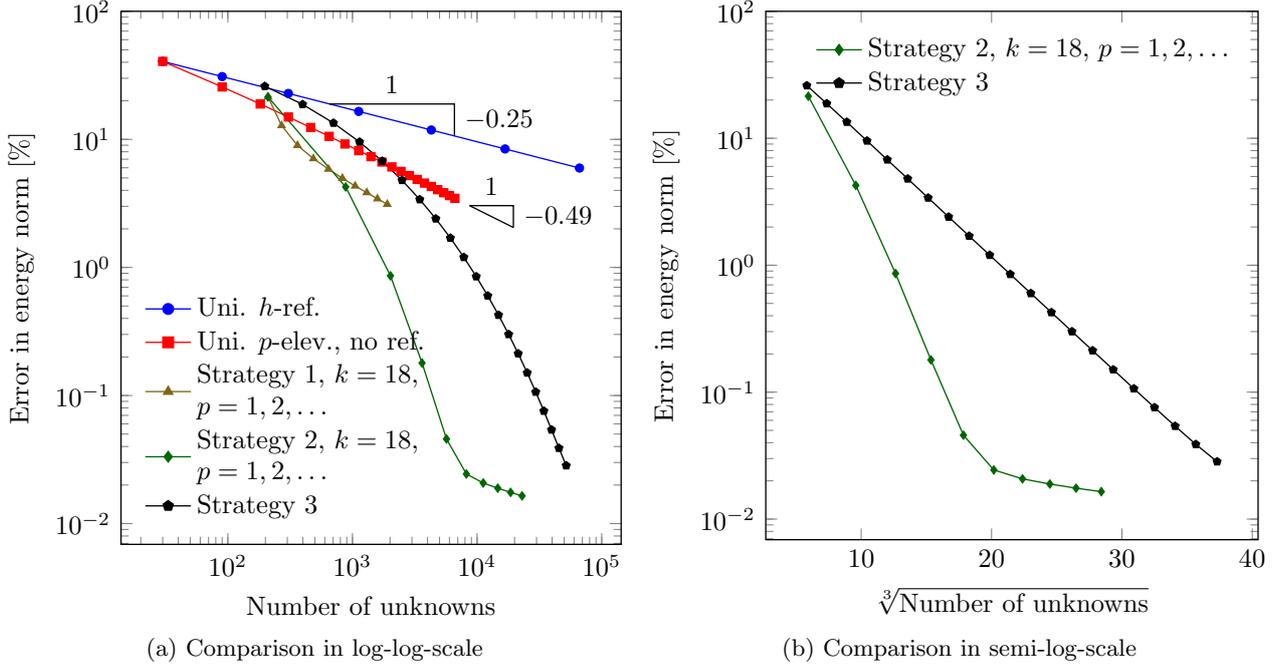


Figure 14: Direct comparison of the accuracy gained by different refinement strategies. Used abbreviations: uni.: uniform, ref.: refinement, elev.: elevation,  $k$ : number of recursive refinements.

**Strategy 2: Uniform  $p$ -elevation with fixed number of high-order overlay meshes** The second refinement approach considered here is the multi-level  $hp$ -approach. Here, the high-order shape functions are moved from the base- to the leaf-elements. This idea is illustrated exemplarily in Figure 15b. As in the previous case, we use an 18-times recursively refined mesh and elevate the approximation order  $p$  successively from one to ten. The resulting convergence curve presented in Figure 14a demonstrates that the use of high-order overlay meshes has a substantial effect on the convergence characteristic. Although the asymptotic convergence is still algebraic, the convergence rates in the pre-asymptotic range are significantly higher. Only when the polynomial order of the shape functions exceeds the optimal value for the current mesh, the convergence rates decrease and tend towards the expected algebraic value of  $\beta \approx \lambda$  in the asymptotic range. However, when using an 18-times refined mesh for the present example, this turning point lies in the range of  $10^{-2}\%$ . Therefore, the accuracy increases by three orders of magnitude in the pre-asymptotic range. This gain in accuracy could not be achieved with any of the previous refinement strategies.

When depicting the results in a log- $\sqrt[3]{\cdot}$ -scaling in Figure 14b, the error decay appears linear until the asymptotic range is entered. This allows to identify the pre-asymptotic convergence as exponential in the form

$$\|e\|_E \leq C \exp\left(\gamma N^\theta\right) \quad \text{with } \theta = 1/3, \quad (17)$$

and  $C$  and  $\gamma$  denoting positive coefficients independent of  $h$  and  $p$  [80]. This matches the theoretical expectations of an  $hp$ -approximation in the presence of a singularity given in e.g. [38, 80, 92, 93].

**Strategy 3: Simultaneous change of  $h$  and  $p$  using a linear  $p$ -distribution** In the second refinement strategy, the convergence rate eventually turns algebraic. This happens because the error contribution

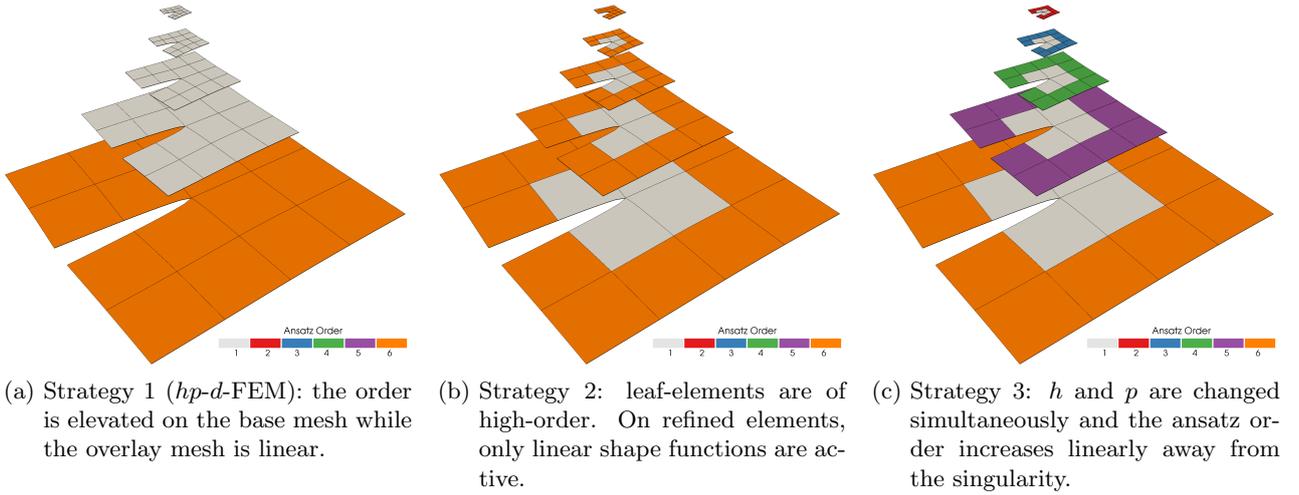


Figure 15: Comparison of  $p$ -distribution of the analyzed refinement strategies with four levels of refinement.

of the elements at the singularity only decays algebraically [37] and therefore eventually dominate the overall convergence. Hence, an exponential convergence in the asymptotic sense requires a decrease of the finest element size  $h$  and a simultaneous elevation of the approximation order  $p$ . This is done in the third refinement strategy considered here. Following the standard idea of an  $hp$ -mesh design presented e.g. in [37, 38], the approximation order  $p$  is increased linearly away from the singularity. An example of such a discretization is depicted in Figure 15c. For the convergence curve presented in Figure 14a, the approximation quality is increased by successively superposing an addition layer of refined elements at the crack tip. Simultaneously, the polynomial order of all other elements is elevated by one. The results depicted in Figure 14 show that this refinement approach yields an exponential convergence characteristic also in the asymptotic range. Interestingly, however, the pre-asymptotic convergence of refinement strategy 2 reaches a better approximation quality than the simultaneous change of  $h$  and  $p$ . This is counter-intuitive and indicates that the chosen linear increase of  $p$  is not optimal yet. Therefore, we expect that using an error-based refinement indicator would further increase the convergence rate. The formulation of such an adaptive scheme is subject to ongoing research.

#### 4.1.3 Runtime comparison

We want to close this comparison of the different refinement strategies by assessing the convergence of the approximation error not only with respect to the number of unknowns but also with respect to the computational time. To this end, Figure 16a depicts the error against the time needed to integrate, assemble and solve the finite element equation system. Thereby, we used a conjugate gradient solver with a simple diagonal pre-conditioning [94]. The results show that the super-algebraic convergence character achieved by refinement strategy 1 and 2 also carries over to the computational performance. In contrast, the asymptotic convergence achieved by the first refinement strategy remains algebraic also when considering the runtime of the simulation.

To assess whether the performance of the method is in agreement with the theoretical complexity, we follow e.g. [95] and estimate that the workload  $W$  of building the stiffness matrix scales cubically

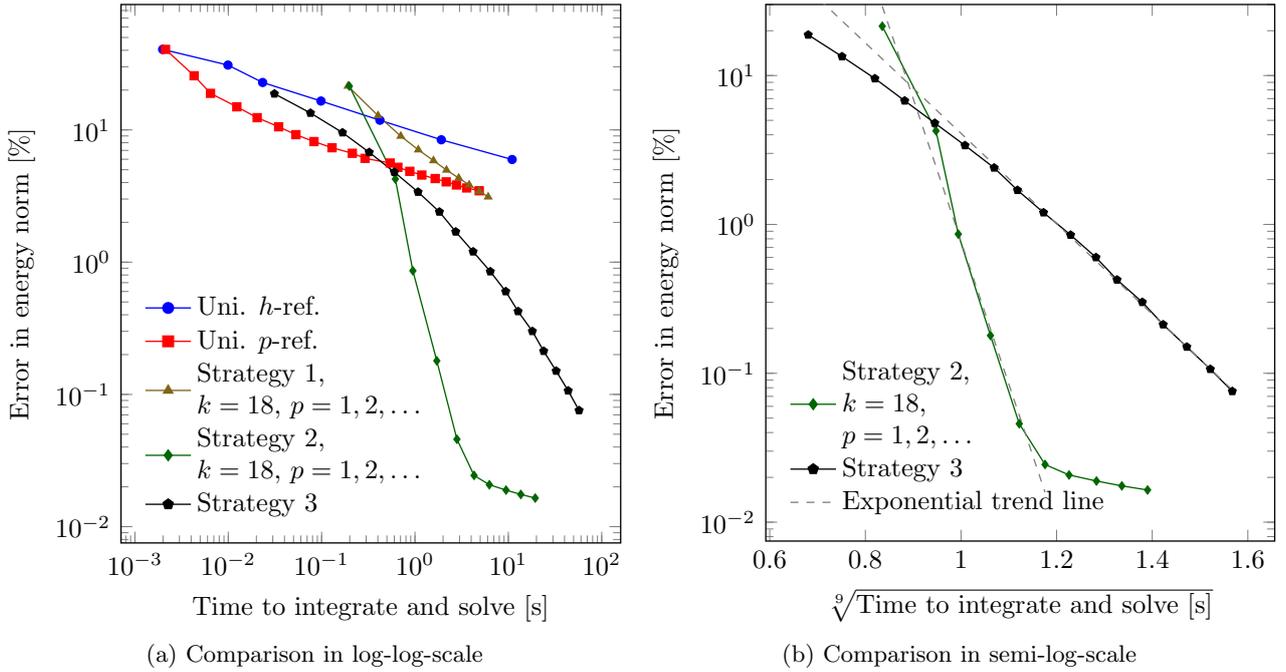


Figure 16: Convergence of approximation error with respect to the time needed to integrate and solve the finite element system. The runtime measurements were performed on a desktop workstation using a single core of an Intel<sup>®</sup> Core<sup>™</sup> i7-4790 CPU and the same finite element code framework.  $k$ : number of recursive refinements.

with the problem size  $N$

$$W = \mathcal{O}(N^3). \quad (18)$$

In combination with the estimate (17), this yields the following bound for the error in terms of the workload

$$\|e\|_E \leq C \exp\left(\omega W^{1/9}\right), \quad (19)$$

with  $C$  and  $\omega$  denoting positive coefficients independent of  $h$  and  $p$ . To compare the numerical performance of the refinement methods to this theoretical estimate, we depicting the results in a log- $\sqrt[9]{N}$  scale in Figure 16b. In this scaling, the pre-asymptotic convergence of strategy 2 and the asymptotic convergence of strategy 3 appear virtually linear. Therefore, we can identify these convergences as exponential with respect to the 9<sup>th</sup>-root of the run time, which is in excellent agreement with estimate (19). This demonstrates that the exponential convergence character of the multi-level  $hp$ -method translates to computational effort in the theoretical correct way.

When comparing the different refinement strategies to each other, the second strategy clearly outperforms the other approaches: Within the pre-asymptotic range, the computation time of strategy 2 increases by a factor of about 20, whereas the approximation error decreases by almost three orders of magnitude. At the same runtime, the other approaches still yield an approximation error of more than 1%.

This analysis shows that a uniform order elevation on a pre-refined multi-level  $hp$ -mesh yields the best approximation quality with respect to both, the number of unknowns and the computational time. Due to this superior performance, this approach is used for the remaining examples.

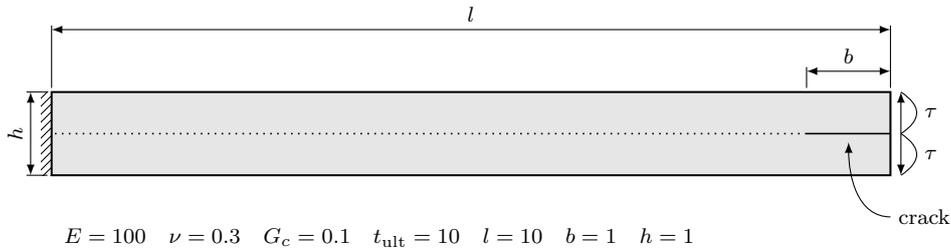


Figure 17: Setup of a double cantilever beam benchmark problem.

## 4.2 Double cantilever beam benchmark

With this example, we demonstrate that the proposed refinement scheme prevents efficiently an unphysical snap-through and snap-back behavior of the load-displacement curve. To this end, the double cantilever beam (DCB) benchmark problem is chosen, which has been studied before in e.g. [18, 19, 21, 22, 24, 26, 28, 29]. The problem is reduced to two dimensions by applying plane strain conditions. The geometry and the material properties are depicted in Figure 17. The two beams are loaded by shear tractions applied to the front face of the two beams as illustrated in Figure 17. The applied shear traction is distributed quadratically over the height of the two beams such that the load is zero at the upper and lower boundaries of the two beams and thus consistent.

The cohesive layer between the two beams is modeled using the exponential cohesive zone law introduced in [65] (see Equation (2) and Figure 1b). The aim of this example is to demonstrate that the proposed refinement scheme works reliably under extreme conditions. For this purpose, the material parameters are chosen extremely brittle. To follow the expected snap-through and snap-back phenomena, which are commonly present for this type of analysis, a dissipation-based arc-length method as proposed in [96] is used.

A first analysis model is composed of 640 linear elements over the beam length and 10 elements over the height of each beam. In Figure 18, the load-displacement curve of the beam tip obtained with this discretization is depicted. The results show that using a sufficiently fine discretization, the curve is smooth and free of numerical artifacts. However, already doubling the element size along the horizontal axis introduces a series of minor snap-through and snap-back effects. Reducing the resolution even further to 160 linear elements results in severe discretization-induced oscillations, which render the result unphysical. Even more, as the crack propagates and the stiffness of the beam decreases, the oscillation cycles tilt more and more horizontal. As a consequence, the number of load steps significantly increases compared to the initially fine discretization model.

These results demonstrate the intrinsic dependency of the cohesive zone model on the discretization. In the following, we study the effects of global  $p$ -refinement and local  $hp$ -refinement. To this end, we compare the load-displacement curves obtained from different discretization models in Figure 19.

We start with a uniform  $p$ -refinement of the model. For this purpose, the discretization is coarsened from 320 to 160 elements along the beam length. To compensate the larger element size, the polynomial order of the shape functions is increased from  $p = 1$  to  $p = 2$ . Thus, the total number of unknowns remains unchanged. However, the numerical stiffness of the system changes as the higher-order elements can better represent the quadratic shear stress. This results in a parallel shift of the load-displacement curve, *cf.* Figure 19a, which in the mean is closer to the sufficiently converged reference solution shown in Figure 19f. Increasing the polynomial degree further to  $p = 3$  reduces the oscillation effect noticeably. Furthermore, the higher order shape functions allow a reduction of the number of elements over the beam height to two elements without compromising the accuracy of the computation. In this way, the number of unknowns significantly reduces with immediate effect on the computational performance. For this reason, the following higher-order computations are performed

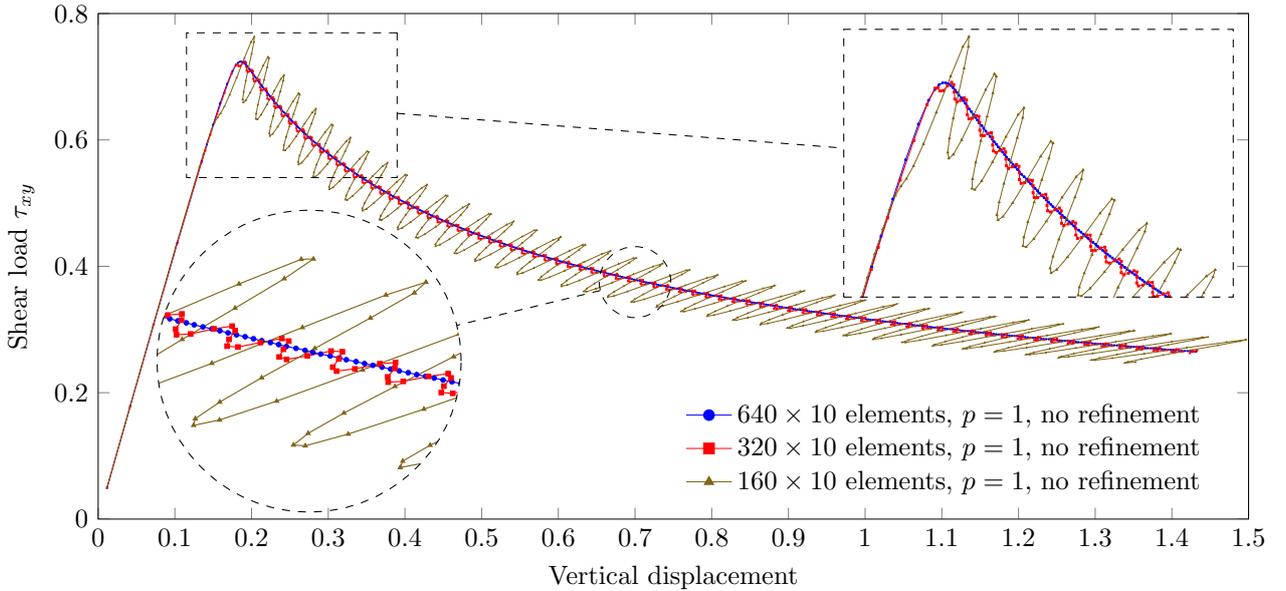


Figure 18: Global load-displacement curve of beam tip

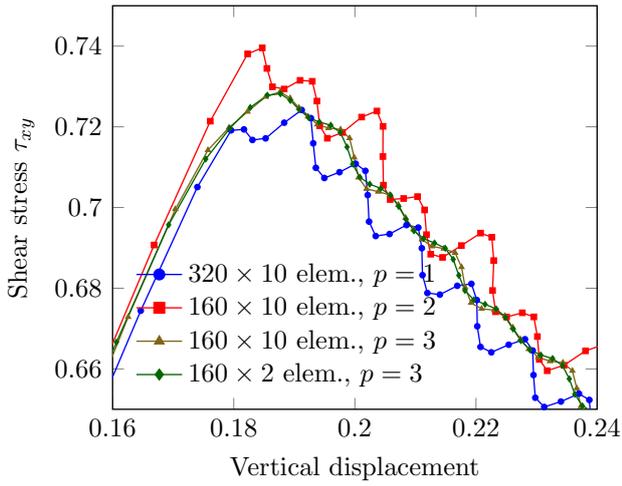
with only two elements over the height of the beam.

In Figure 19b, we show that a further order elevation significantly reduces the amplitude of the unphysical oscillations. Similar positive effects of higher order discretizations on the stability of the solution scheme have been reported in the context of isogeometric analysis in e.g. [28]. However, this positive effect has its limits as shown in Figure 19c: further coarsening of the mesh from 160 to 80 elements reveals the dependency of the solution behavior on the element size even for a higher approximation order. This demonstrates that an accurate and efficient representation of the crack propagation needs both, a small element size and high-order shape functions. However, this refined discretization is only needed in the direct vicinity of the failure zone, i.e. the crack tip. All other parts of the analysis domain may fully profit from larger high-order elements providing an accurate stress representation.

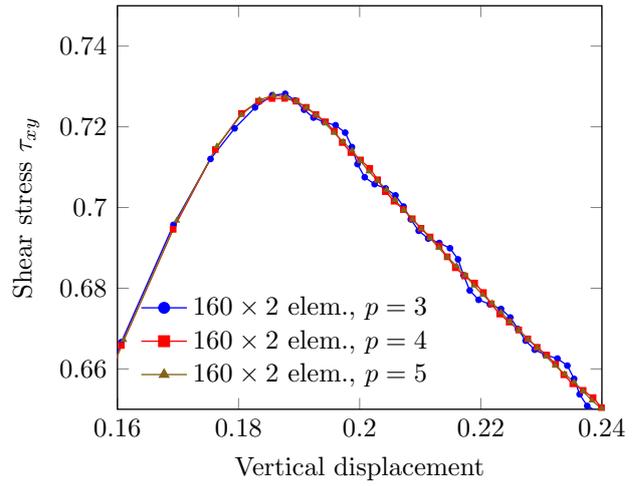
For this reason, we use the proposed  $hp$ -adaptive scheme to refine the mesh locally in the vicinity of the crack tip. As outlined in Section 3.2.2, the gap opening computed in the last load step serves as a refinement criterion. We refine all elements along the crack interface whose deformation results in a normal gap opening that is larger than  $g_1 = 10^{-4}$  and smaller than  $g_2 = 5 \cdot 10^{-2}$ . The refinement is performed up to a pre-defined refinement depth. If the gap opening is larger than  $g_2$  along the complete edge of an element, the element is coarsened again until the original base mesh is reached.

In the study, depicted in Figure 19d, a base discretization of  $160 \times 2$  quadratic elements is used. In combination with one level of refinement, the element size at the crack tip thus corresponds to a discretization using  $320 \times 2$  quadratic elements. However, when refining by only linear overlay elements ( $hp$ - $d$ -refinement, Figure 7a) reveals no visible improvement compared to the results obtained from the coarse, unrefined discretization. If we instead refine the base mesh by superposing high-order overlay elements (multi-level  $hp$ -refinement, Figure 7c), the magnitude of the oscillation reduces to the same size obtained by the globally fine discretization of  $320 \times 2$  quadratic elements. This analysis shows that the correct capturing of the propagating crack front requires high-order shape functions on the *smallest* support, which can be achieved using the multi-level  $hp$ -formulation. This confirms the expectations of Section 3.3.

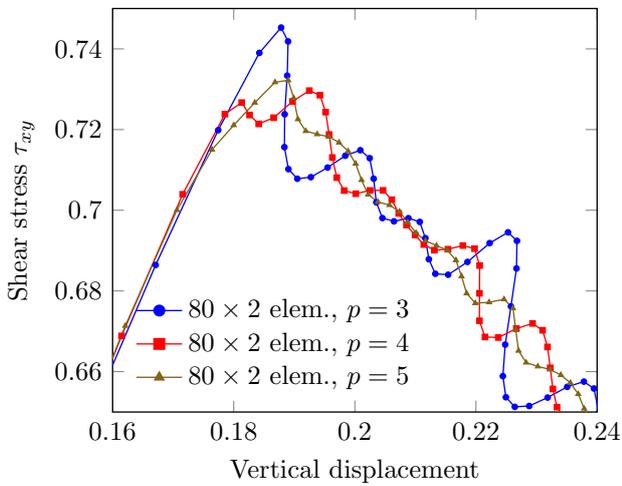
The results obtained by the global and the local multi-level  $hp$ -refinement only differ in a slight parallel shift of the curves. This is again due to the different stiffness of the numerical systems



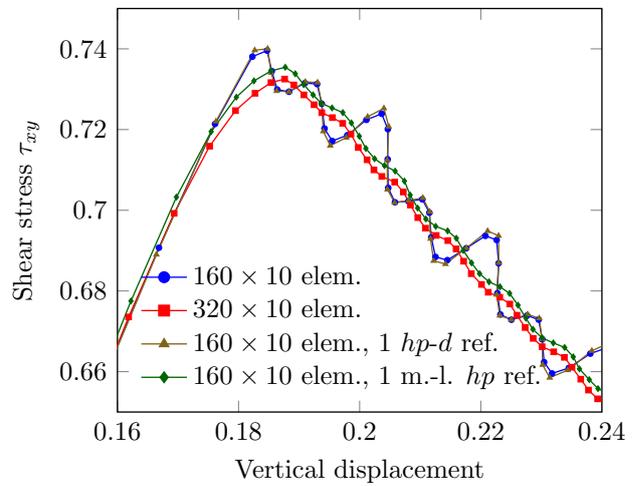
(a) Global  $p$ -refinement on fine discretization (part 1)



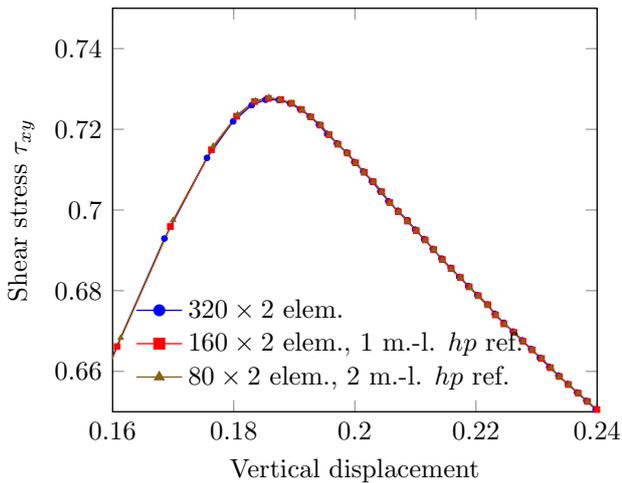
(b) Global  $p$ -refinement on fine discretization (part 2)



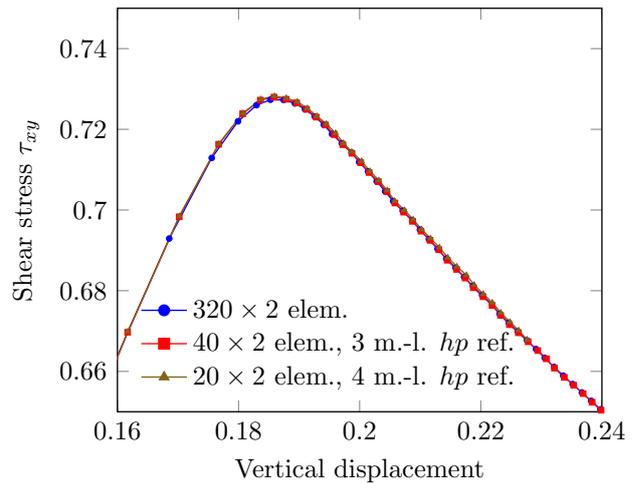
(c) Global  $p$ -refinement coarse discretization



(d) Local  $hp$ -d and multi-level  $hp$ -refinement for  $p = 2$



(e) Local multi-level  $hp$ -refinement for  $p = 3$  (part 1)



(f) Local multi-level  $hp$ -refinement for  $p = 3$  (part 2)

Figure 19: Comparison of load curves at limit point for different discretizations (used abbreviations: elem. = elements, ref. = refinement(s), m.-l.  $hp$  = multi-level  $hp$ ).

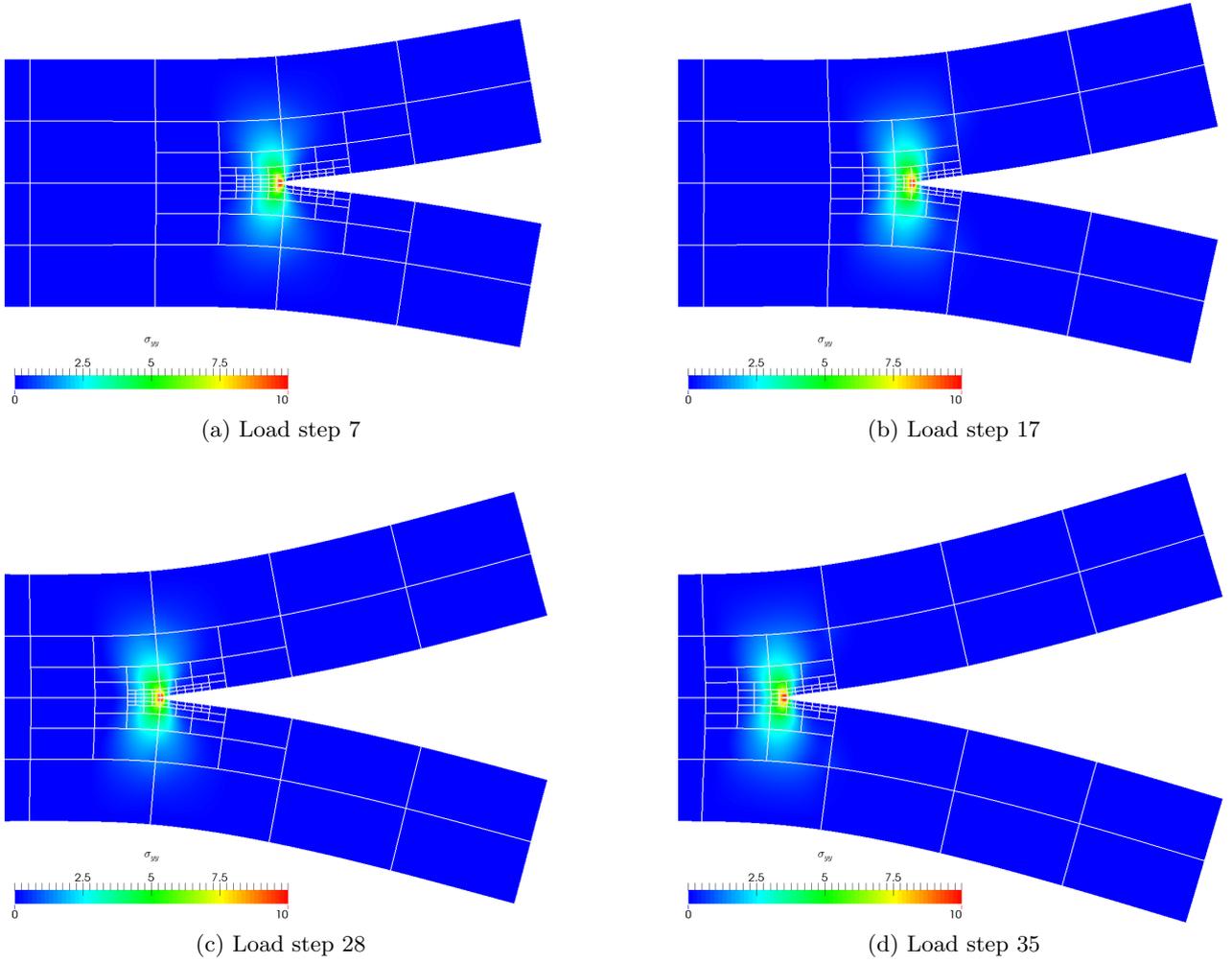


Figure 20: Stress distribution in deformed beam at different load steps ( $20 \times 2$  elements,  $p = 3, 4$  levels of multi-level  $hp$ -refinement)

caused by the inability of the quadratic elements to correctly capture the shear stress in the beam. For this reason, the study is repeated using cubic shape functions. The results depicted in Figure 19e demonstrate an excellent agreement with the reference solution based on a globally refined mesh.

The further results shown in Figures 19e and f demonstrate that the same quality of results can be achieved by using even coarser base meshes when increasing the refinement depth accordingly. Even using only 20 elements over the beam length and four levels of refinements shows a virtually identical load curve.

In Figure 20, we exemplarily depict the stress concentration ahead of the propagating crack tip for four different load steps. The computation is performed on a base mesh of  $20 \times 2$  elements, which is refined four times towards the crack tip. The transition of fracture process zone through the elements takes place without oscillations in the stress state. As soon as the crack has traveled completely through an element domain, this element is coarsened again by simply reducing the depth of the refinement tree.

In Table 1, the maximum number of unknowns are listed for various discretizations with comparable accuracy. The comparison shows that already the use of higher-order bulk elements is superior to

No. of elements	$p$	No. of refinements	Max. no. of dofs	Relative runtime [%]
$640 \times 10$	1	0	28,204	122.1
$320 \times 2$	3	0	26,908	100.0
$160 \times 2$	3	1	14,428	41.1
$80 \times 2$	3	2	8,236	24.5
$40 \times 2$	3	3	5,188	20.1
$20 \times 2$	3	4	3,388	19.4

Table 1: Comparison of the number of unknowns and the computational time using the adaptive refinement. The runtime measurements were performed on a desktop workstation using a single core of an Intel<sup>®</sup> Core<sup>™</sup> i7-4790 CPU and the same finite element code framework.  $k$ : number of recursive refinements.

a fine, low-order mesh. In the present case, the dominating bending mode results in a quadratic distribution of the shear stress over the beam height. Thus, cubic elements are optimal in this example. Other applications, however, might benefit from even higher order shape functions. Further, the results demonstrate that—compared to a refinement along the full crack path—the use of a dynamic local mesh refinement yields a significant reduction of the necessary degrees of freedom and the computational time without degrading the result quality. For the present case, using the suggested local refinement approach decreases the runtime by a factor of five. The expected gain in performance can even be higher when increasing the length of the beams while keeping the length of the cohesive zone unchanged.

### 4.3 Delamination in three-dimensional models

The previous section revealed the potential of the multi-level  $hp$ -formulation for delamination problems with regard to computational efficiency. In particular, we demonstrated that the proposed method easily overcomes unphysical oscillations, which are inherent to cohesive crack propagation with standard discretizations. Based on the previous model, we consider in the following an extension to three dimensions. The purpose of this example is to study the three-dimensional stress state along the crack front. To this end, the laminated plates depicted in Figure 21 are considered.

Again, we use the Xu-Needleman traction separation law to model the cohesive crack propagation.

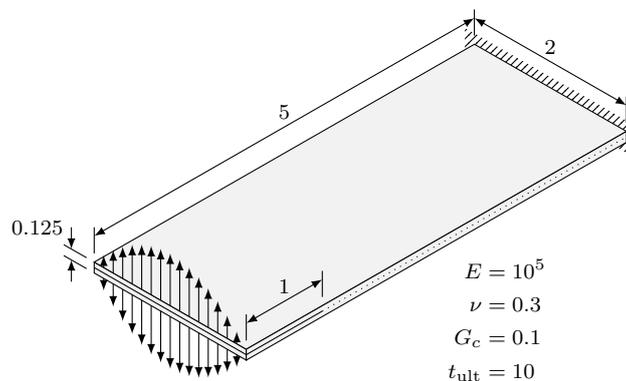


Figure 21: Double cantilever plate model

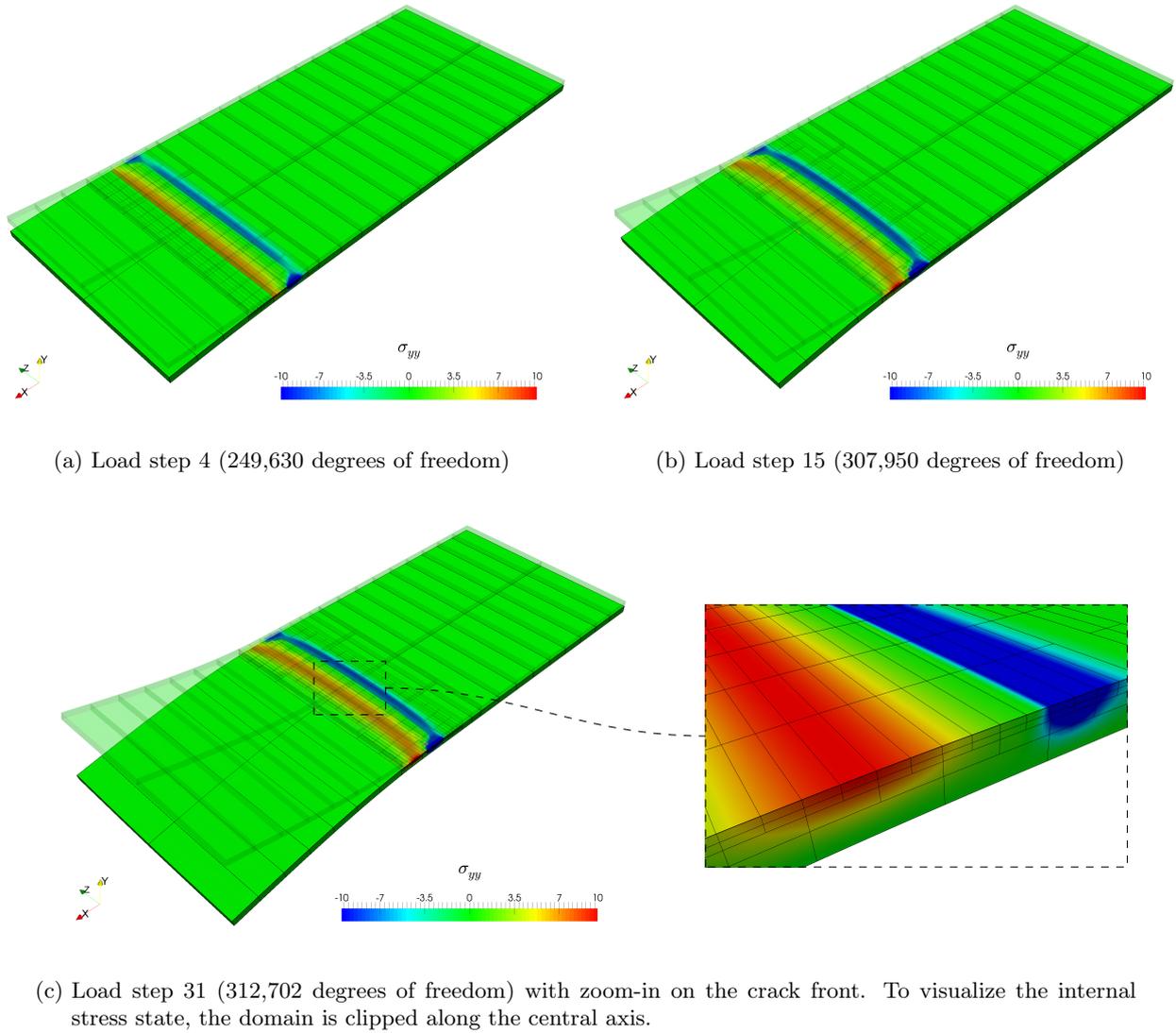


Figure 22: Stress distribution in deflected beam for different load steps ( $1 \times 2 \times 20$  elements,  $p = 4$ , three levels of refinement).

In analogy to the previously discussed two-dimensional example, the loading is applied as a shear traction on the front faces of the two plates. To ensure consistent loading conditions, the load is applied in a bi-quadratic distribution that is zero on all four boundaries of the front faces.

Common discretization-approaches for the considered thin plates employ dimensionally reduced shell-elements, which are based on assumptions on the through-thickness displacement. A comprehensive review of this technology is given in e.g. [97]. In the present case, however, the crack front yields a complex stress state that cannot be incorporated easily in such models. For this reason, this element technology is not used here. Instead, the geometry is discretized with thin, volumetric, three-dimensional hexahedral elements of higher order. In [98], it was shown that such hierarchic high-order shape functions qualify well for the analysis of thin shell models. In particular, the high-order shape functions alleviate locking phenomena, which is a common issue for standard low-order shell elements. This allows for coarse, thin elements with a high aspect ratio.

Following this approach, each plate is discretized using  $1 \times 2 \times 20$  elements of order  $p = 4$ . This

base mesh is refined locally around the crack front to resolve the cohesive zone accurately. As in the two-dimensional case, the gap opening serves as the criterion to decide about multi-level  $hp$ -refinement. The range where to refine is selected as  $g_1 = 5 \cdot 10^{-6}$  and  $g_2 = 10^{-2}$ . Using three levels of adaptive refinement, this discretization yields up to 340,000 degrees of freedom. In contrast, a global refinement with a comparable element size would result in more than 8,200,000 unknowns, which demonstrates the significant reduction of computational effort gained by the local refinement.

As depicted in Figure 22, the  $hp$ -approach allows to resolve the cohesive zone ahead of the crack front while it is moving through the domain. In particular, the curvature of the crack front is captured automatically by the refinement. Furthermore, the refinement stays local to the crack as the scheme allows for arbitrary hanging nodes. It sharply resolves both the cohesive tension forces as well as the pressure reaction force. A zoom-in on the stress state is shown in Figure 22c. It demonstrates that at the crack front, the stress varies rapidly in the in-plane direction and through the thickness of the plates. The complexity of this stress state justifies the use of refined volumetric elements for the computation of the delamination process.

The depicted results also indicate a concentration of compressive stress at the boundary. This phenomena is caused by the tilting of the cross-sections at these points yielding a localized contact of the two plates on the edges. Even the refined mesh is too coarse to capture the resulting singular-like stress-state. An automated refinement allowing the accurate resolution of this feature would demand for an error-based adaptivity scheme. The formulation of such an automated approach is subject to current research.

#### 4.4 Combination of multi-level $hp$ -adaptivity and the finite cell method

This final example demonstrates the potential of the chosen approach in the context of complex geometric domains. To this end, we have chosen a perforated double cantilever beam as introduced in [96]. The geometry and material properties are depicted Figure 23. As in Example 4.2, we use the Xu-Needleman traction-separation law and load the structure by a quadratically distributed shear force applied to the front faces of the two beams.

Due to the perforation, the delamination of the considered structure yields a complex load-displacement curve characterized by *physically-correct* snap-back effects. Thus, the simulation demands for a sophisticated path-following technique. As shown in [96], a dissipation-based criterion is well suited for this need and will thus be used here as well.

In the original work of Verhoosel *et al.* [96], the geometry was discretized using triangle elements. The resolution of the fine-scale geometrical features demands for a small element size, which is accompanied by a high number of unknowns. To circumvent this problem, the finite cell method (FCM), introduced in [99, 100], is used to model the perforation in this example.

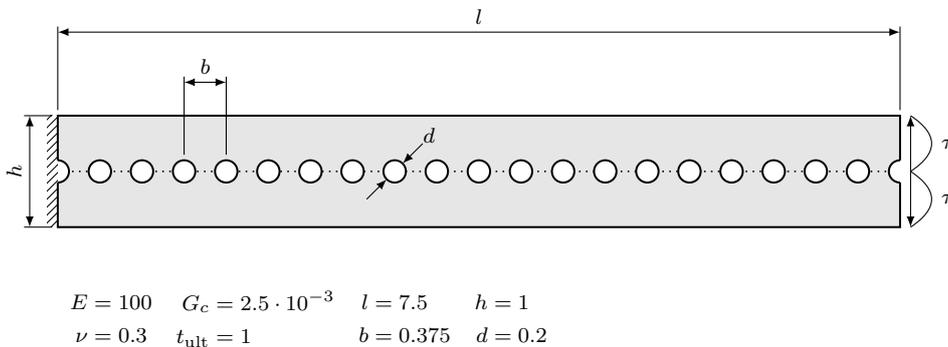


Figure 23: Setup of the perforated double cantilever beam benchmark following [96].

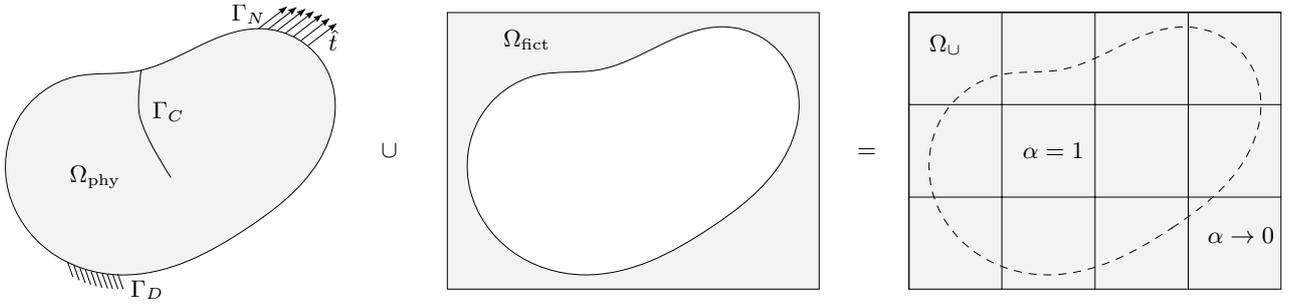


Figure 24: Principle idea of the finite cell method (FCM) following [99]: the complex physical domain  $\Omega_{\text{phy}}$  is extended by a fictitious domain  $\Omega_{\text{fict}}$  such that their union  $\Omega_{\cup}$  yields a much simpler geometry. The original problem is recovered using the indicator function  $\alpha$ .

The core principle of this method is to combine the advantageous approximation properties of higher-order elements with the idea of fictitious domain approaches. To this end, the complex physical domain  $\Omega_{\text{phy}}$  is embedded in a fictitious domain  $\Omega_{\text{fict}}$  such that their union  $\Omega_{\cup} = \Omega_{\text{phys}} \cup \Omega_{\text{fict}}$  yields a simple geometry *cf.* Figure 24. The domain  $\Omega_{\cup}$  is discretized with high-order elements on a Cartesian grid. Using an indicator function

$$\alpha(\mathbf{x}) = \begin{cases} 1 & \forall \mathbf{x} \in \Omega_{\text{phy}} \\ \varepsilon \rightarrow 0 & \forall \mathbf{x} \notin \Omega_{\text{phy}}, \end{cases} \quad (20)$$

the original problem can be recovered on the integration level by scaling the integrands of the weak form.

This simple idea has proven its excellent applicability in various contexts, such as linear elasticity [99, 100], geometrically nonlinear continuum mechanics [88, 101], contact mechanics [102], bio-medical engineering [103, 104], thin-walled structures [105], design-through analysis and isogeometric analysis [72, 87, 106–108], and fluid-structure interaction [109]. A comprehensive review of the method and its recent extensions is given in [110]. An open-source MATLAB-toolbox presented in [111] offers an easy start into this research field.

In the present example, we use the finite cell method to simulate the delamination of the perforated structure depicted in Figure 23. To this end, the weak form introduced in Section 2 is reformulated following the finite cell idea:

$$\text{Find } \mathbf{u} \in \mathcal{V}, \text{ such that} \\ \int_{\Omega_{\cup}} \delta \boldsymbol{\varepsilon} : \alpha \boldsymbol{\sigma} \, d\Omega + \int_{\Gamma_C} \delta \llbracket \mathbf{u} \rrbracket \cdot \alpha \mathbf{t}(\llbracket \mathbf{u} \rrbracket) \, d\Gamma = \int_{\Omega_{\cup}} \delta \mathbf{u} \cdot \alpha \hat{\mathbf{b}} \, d\Omega + \int_{\Gamma_N} \delta \mathbf{u} \cdot \hat{\mathbf{t}} \, d\Gamma \quad \forall \delta \mathbf{u} \in \mathcal{V}_0, \quad (21)$$

with  $\mathcal{V}$  and  $\mathcal{V}_0$  again denoting the function spaces defined in (2.2). Note that also the left hand side integral over  $\Gamma_C$  is augmented by the indicator function  $\alpha$ . This allows for a traction-free delamination within the fictitious domain. With this change of the problem, the two beams depicted in Figure 23 are discretized using  $20 \times 2$  hexahedral elements in which the perforation is treated as fictitious domain. The geometry of the perforation is resolved on the integration level using a quadtree-based sub-cell integration scheme [100, 110], which densifies integration points along the domain interfaces to capture properly the true integration domain  $\Omega_{\text{phys}}$ . Such a simple discretization allows to reduce significantly the number of unknowns compared to a boundary-conforming finite element approach. However, the coarse base mesh is not sufficient to capture the complex stress patterns emerging around the holes. Following the previous examples, the discretization is, therefore, refined locally

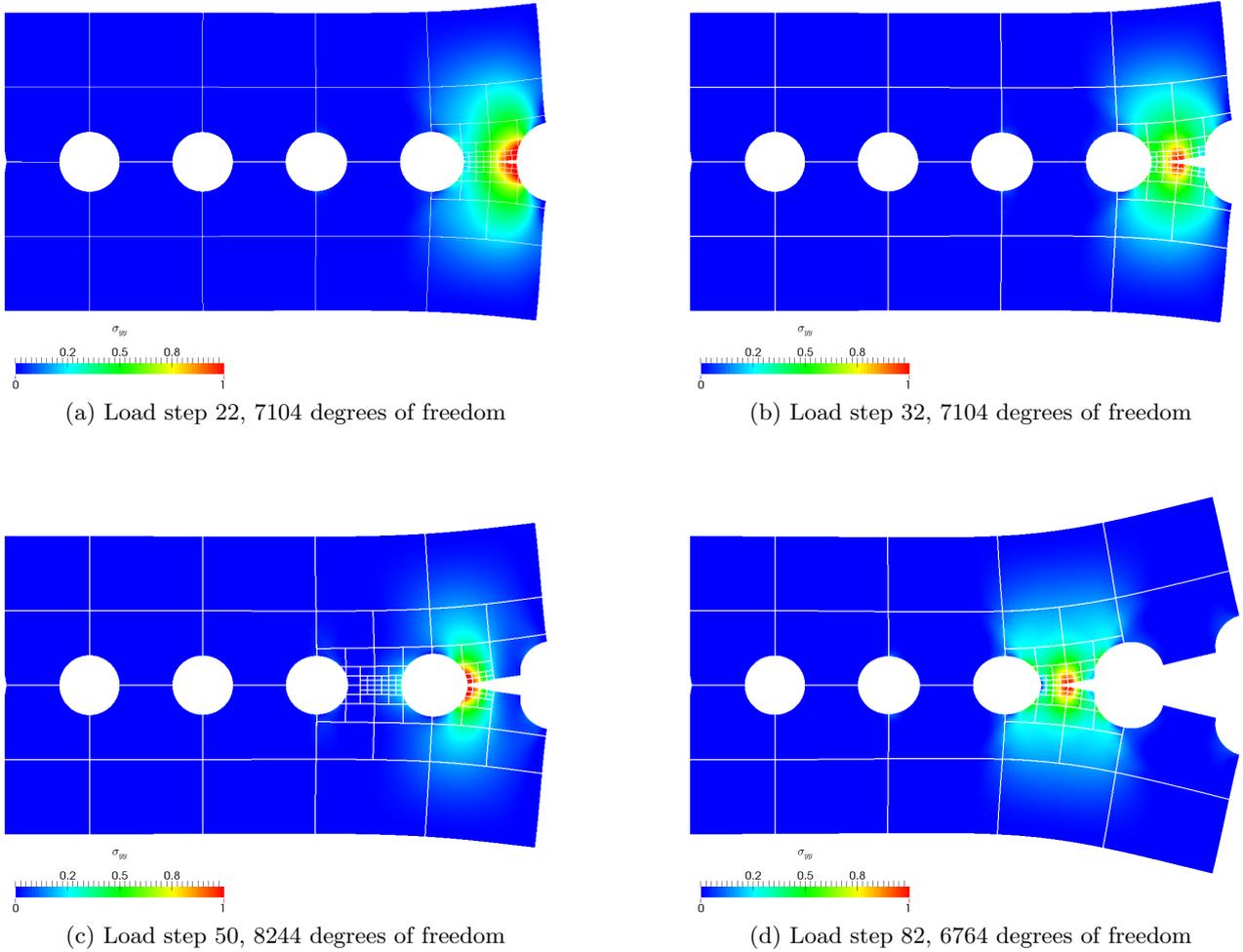


Figure 25: Stress distribution on the deformed structure at different load steps,  $20 \times 2$  elements,  $p = 5$ , 4 levels of multi-level  $hp$ -refinement, max. 8564 degrees of freedom throughout the analysis.

around the moving crack tip using four levels of multi-level  $hp$ -refinement. The gap-range in which we select the elements for refinement is set to  $g_1 = 2.5 \cdot 10^{-6}$  and  $g_2 = 2.5 \cdot 10^{-3}$ .

In Figure 25, we depict the stress concentration moving ahead of the crack-opening, which is computed using the dynamically changing multi-level  $hp$ -mesh. In Figure 26, the load-displacement curve of the problem is depicted, which is characterized by a number of complex snap-back effects. It is worth to notice that the fictitious domain extended problem was solved stably and reliably using the dissipation-based arc-length method presented in [96]. A comparison of the multi-level  $hp$ -derived load-displacement curve with a result obtained using a uniformly refined mesh shows excellent agreement at one order of magnitude less degrees of freedom. The quality of results presented for this problem clearly indicate the potential of the multi-level  $hp$ -refinement in combination with the finite cell method for propagation problems with complex domain geometry.

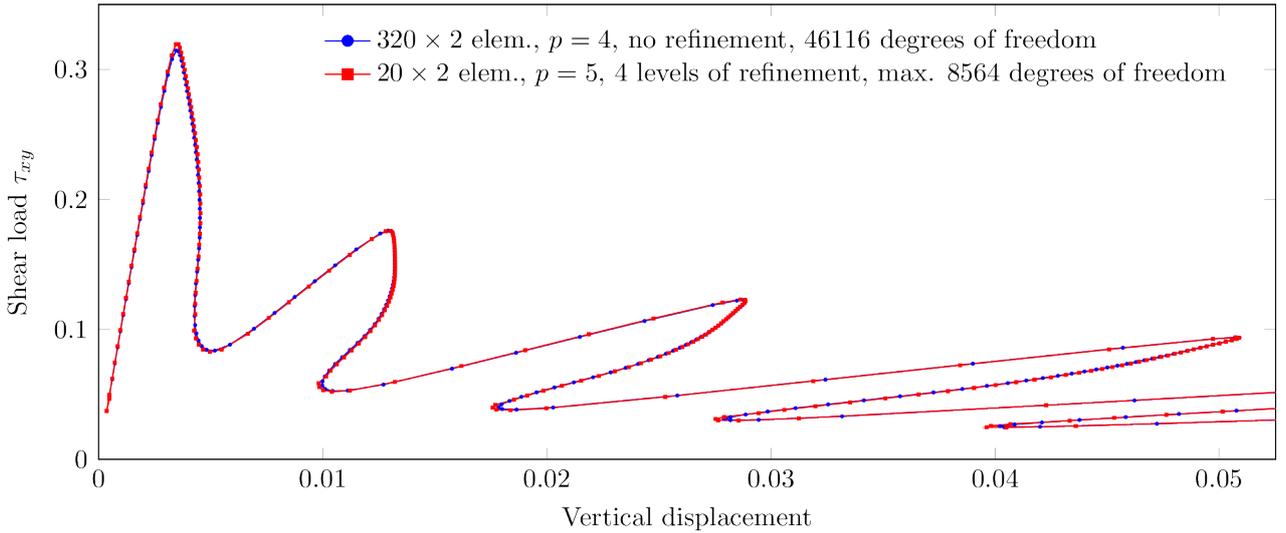


Figure 26: Global load-displacement curve of beam tip.

## 5 Conclusion

In the work presented here, we demonstrated that the approximation quality of propagating cohesive delamination can be increased efficiently by performing an  $hp$ -mesh refinement in the vicinity of the current crack tip position. As the crack is propagating through the domain during the simulation, the discretization has to be adapted in every load step. To this end, we used the recently introduced multi-level  $hp$ -version of the finite element method, in which a coarse base discretization was refined by superposing a fine overlay mesh in the domain of interest. This approach allows to dynamically change the discretization during the simulation by simply moving the position of the overlay mesh while keeping the base mesh unchanged.

In the context of an analytical benchmark, we demonstrated that—even in the presence of stress singularities—the suggested  $hp$ -method yields an exponential convergence of the approximation error both, with respect to the degrees of freedom and the simulation runtime. Secondly, we systematically analyzed the influence of  $h$ -,  $p$ - and  $hp$ -refinement on the result quality in the context of the double cantilever beam (DCB) benchmark problem. We could show that the dynamic discretization capabilities of the multi-level  $hp$ -approach are well suited for a non-uniform  $hp$ -mesh refinement that follows the propagating crack tip: the global deformation of the structure is captured efficiently using a coarse, high-order base mesh, while the complex stress state in the fracture process zone is resolved on a fine overlay mesh. In this way, artificial, discretization induced oscillations in the load-displacement curve—which are a well-known problem in the simulation of cohesive crack growth—can be prevented effectively. By following the propagating crack tip with the overlay mesh, the refinement can stay local to the crack and no *a priori* mesh-refinement along the full crack path is required. Our studies showed that this significantly reduces the number of unknowns and with it computational time needed for the simulation. The same positive effects were observed in the simulation of three-dimensional delamination. We showed that the refinement scheme can follow the curved crack front and resolves the complex, three-dimensional stress state at the crack front with high accuracy. In a third example, we combined the presented approach successfully with the finite cell method to simulate delamination within complex geometries. The local refinement around the crack tip achieved the same result quality as a refinement along the full crack path using one order of magnitude less degrees of freedom.

Based on the presented results, we conclude that the combination of the cohesive zone approach,

the multi-level  $hp$ -refinement scheme and the finite cell method yields a highly effective scheme for the accurate simulation of crack propagation in complex domains.

## Acknowledgements

The first and the last author gratefully acknowledge the financial support of the German Research Foundation (DFG) under Grant RA 624/22-1. N. Zander gratefully acknowledges support from TU Delft, Aerospace Engineering, and from the TUM Graduate School during his three months stay at Delft.

## References

- [1] D. S. Dugdale, “Yielding of steel sheets containing slits,” *Journal of the Mechanics and Physics of Solids*, vol. 8, no. 2, pp. 100–104, 1960.
- [2] G. I. Barenblatt, “The Mathematical Theory of Equilibrium Cracks in Brittle Fracture,” in *Advances in Applied Mechanics* (H. Dryden, T. von Kármán, G. Kuerti, F. H. van den Dungen, and L. Howarth, eds.), vol. 7, pp. 55–129, Elsevier, 1962.
- [3] A. Hillerborg, M. Modéer, and P. E. Petersson, “Analysis of crack formation and crack growth in concrete by means of fracture mechanics and finite elements,” *Cement and Concrete Research*, vol. 6, no. 6, pp. 773–781, 1976.
- [4] A. Needleman, “A Continuum Model for Void Nucleation by Inclusion Debonding,” *Journal of Applied Mechanics*, vol. 54, no. 3, pp. 525–531, 1987.
- [5] A. Needleman, “An analysis of decohesion along an imperfect interface,” *International Journal of Fracture*, vol. 42, no. 1, pp. 21–40, 1990.
- [6] A. Needleman, “An analysis of tensile decohesion along an interface,” *Journal of the Mechanics and Physics of Solids*, vol. 38, no. 3, pp. 289–324, 1990.
- [7] V. Tvergaard and J. W. Hutchinson, “The relation between crack growth resistance and fracture process parameters in elastic-plastic solids,” *Journal of the Mechanics and Physics of Solids*, vol. 40, no. 6, pp. 1377–1397, 1992.
- [8] X. P. Xu and A. Needleman, “Numerical simulations of fast crack growth in brittle solids,” *Journal of the Mechanics and Physics of Solids*, vol. 42, no. 9, pp. 1397–1434, 1994.
- [9] A. Pandolfi, P. R. Guduru, M. Ortiz, and A. J. Rosakis, “Three dimensional cohesive-element analysis and experiments of dynamic fracture in C300 steel,” *International Journal of Solids and Structures*, vol. 37, no. 27, pp. 3733–3760, 2000.
- [10] G. Ruiz, A. Pandolfi, and M. Ortiz, “Three-dimensional cohesive modeling of dynamic mixed-mode fracture,” *International Journal for Numerical Methods in Engineering*, vol. 52, no. 1-2, pp. 97–120, 2001.
- [11] J. C. J. Schellekens and R. de Borst, “A non-linear finite element approach for the analysis of mode-I free edge delamination in composites,” *International Journal of Solids and Structures*, vol. 30, no. 9, pp. 1239–1253, 1993.
- [12] R. de Borst, J. J. Remmers, and A. Needleman, “Computational aspects of cohesive zone models,” in *Advanced fracture mechanics for life and safety assessments*, (Stockholm), 2004.

- [13] R. de Borst and J. J. C. Remmers, “Computational modelling of delamination,” *Composites Science and Technology*, vol. 66, no. 6, pp. 713–722, 2006.
- [14] P. H. Geubelle and J. S. Baylor, “Impact-induced delamination of composites: a 2D simulation,” *Composites Part B: Engineering*, vol. 29, no. 5, pp. 589–602, 1998.
- [15] J. Planas, M. Elices, G. V. Guinea, F. J. Gómez, D. A. Cendón, and I. Arbilla, “Generalizations and specializations of cohesive crack models,” *Engineering Fracture Mechanics*, vol. 70, no. 14, pp. 1759–1776, 2003.
- [16] M. Samimi, J. A. W. van Dommelen, and M. G. D. Geers, “A self-adaptive finite element approach for simulation of mixed-mode delamination using cohesive zone models,” *Engineering Fracture Mechanics*, vol. 78, no. 10, pp. 2202–2219, 2011.
- [17] Y. Mi, M. A. Crisfield, G. A. O. Davies, and H. B. Hellweg, “Progressive Delamination Using Interface Elements,” *Journal of Composite Materials*, vol. 32, no. 14, pp. 1246–1272, 1998.
- [18] G. Alfano and M. A. Crisfield, “Finite element interface models for the delamination analysis of laminated composites: mechanical and computational issues,” *International Journal for Numerical Methods in Engineering*, vol. 50, no. 7, pp. 1701–1736, 2001.
- [19] M. A. Crisfield and G. Alfano, “Adaptive hierarchical enrichment for delamination fracture using a decohesive zone model,” *International Journal for Numerical Methods in Engineering*, vol. 54, no. 9, pp. 1369–1390, 2002.
- [20] V. Tomar, J. Zhai, and M. Zhou, “Bounds for element size in a variable stiffness cohesive finite element model,” *International Journal for Numerical Methods in Engineering*, vol. 61, no. 11, pp. 1894–1920, 2004.
- [21] A. Turon, C. G. Dávila, P. P. Camanho, and J. Costa, “An engineering solution for mesh size effects in the simulation of delamination using cohesive zone models,” *Engineering Fracture Mechanics*, vol. 74, no. 10, pp. 1665–1682, 2007.
- [22] P. W. Harper and S. R. Hallett, “Cohesive zone length in numerical simulations of composite delamination,” *Engineering Fracture Mechanics*, vol. 75, no. 16, pp. 4774–4792, 2008.
- [23] F. H. Hermes, *Process zone and cohesive element size in numerical simulations of delamination in bi-layers*. Master’s thesis, Technische Universiteit Eindhoven, Eindhoven, 2010.
- [24] I. Guiamatsia, J. K. Ankersen, G. A. O. Davies, and L. Iannucci, “Decohesion finite element with enriched basis functions for delamination,” *Composites Science and Technology*, vol. 69, no. 15–16, pp. 2616–2624, 2009.
- [25] M. Samimi, J. A. W. van Dommelen, and M. G. D. Geers, “An enriched cohesive zone model for delamination in brittle interfaces,” *International Journal for Numerical Methods in Engineering*, vol. 80, no. 5, pp. 609–630, 2009.
- [26] M. Samimi, J. A. W. van Dommelen, and M. G. D. Geers, “A three-dimensional self-adaptive cohesive zone model for interfacial delamination,” *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 49–52, pp. 3540–3553, 2011.
- [27] V. P. Nguyen and H. Nguyen-Xuan, “High-order B-splines based finite elements for delamination analysis of laminated composites,” *Composite Structures*, vol. 102, pp. 261–275, 2013.

- [28] R. Dimitri, L. D. Lorenzis, P. Wriggers, and G. Zavarise, “NURBS- and T-spline-based isogeometric cohesive zone modeling of interface debonding,” *Computational Mechanics*, vol. 54, no. 2, pp. 369–388, 2014.
- [29] Y. Wang and H. Waisman, “Progressive delamination analysis of composite materials using XFEM and a discrete damage zone model,” *Computational Mechanics*, vol. 55, no. 1, pp. 1–26, 2015.
- [30] C. V. Verhoosel, M. A. Scott, R. de Borst, and T. J. R. Hughes, “An isogeometric approach to cohesive zone modeling,” *International Journal for Numerical Methods in Engineering*, vol. 87, no. 1-5, pp. 336–360, 2011.
- [31] V. P. Nguyen, P. Kerfriden, and S. P. A. Bordas, “Two- and three-dimensional isogeometric cohesive elements for composite delamination analysis,” *Composites Part B: Engineering*, vol. 60, pp. 193–212, 2014.
- [32] S. Hosseini, *Numerical simulation of damage mechanisms in composite materials*. PhD thesis, Technische Universiteit Eindhoven, Eindhoven, 2014.
- [33] S. Hosseini, J. J. Remmers, C. V. Verhoosel, and R. de Borst, “Propagation of delamination in composite materials with isogeometric continuum shell elements,” *International Journal for Numerical Methods in Engineering*, vol. 102, no. 3-4, pp. 159–179, 2015.
- [34] K. Park, G. H. Paulino, W. Celes, and R. Espinha, “Adaptive mesh refinement and coarsening for cohesive zone modeling of dynamic fracture,” *International Journal for Numerical Methods in Engineering*, vol. 92, no. 1, pp. 1–35, 2012.
- [35] A. R. Khoei, H. Moslemi, and M. Sharifi, “Three-dimensional cohesive fracture modeling of non-planar crack growth using adaptive FE technique,” *International Journal of Solids and Structures*, vol. 49, no. 17, pp. 2334–2348, 2012.
- [36] W. Gui and I. Babuška, “The h, p and h-p versions of the finite element method in 1 dimension Part I: The error analysis of the p-version,” *Numerische Mathematik*, vol. 49, no. 6, pp. 577–612, 1986.
- [37] W. Gui and I. Babuška, “The h, p and h-p versions of the finite element method in 1 dimension Part II: The error analysis of the h-and h-p versions,” *Numerische Mathematik*, vol. 49, no. 6, pp. 613–657, 1986.
- [38] C. Schwab, *p- and hp-Finite Element Methods: Theory and Applications in Solid and Fluid Mechanics*. Numerical mathematics and scientific computation, Oxford: Oxford University Press, 1998.
- [39] L. Demkowicz, *Computing with hp-adaptive finite elements, Vol. 1: One and Two Dimensional Elliptic and Maxwell Problems*. Applied mathematics and nonlinear science series, Boca Raton: Chapman & Hall/CRC, 2007.
- [40] S. M. Schnepf and T. Weiland, “Efficient large scale electromagnetic simulations using dynamically adapted meshes with the discontinuous Galerkin method,” *Journal of Computational and Applied Mathematics*, vol. 236, no. 18, pp. 4909–4924, 2012.
- [41] L. Korosus and P. Šolín, “An adaptive hp-DG method with dynamically-changing meshes for non-stationary compressible Euler equations,” *Computing*, vol. 95, no. 1, pp. 425–444, 2012.

- [42] P. Šolín and L. Korous, “Space-time adaptive hp-FEM for problems with traveling sharp fronts,” *Computing*, vol. 95, no. 1, pp. 709–722, 2012.
- [43] G. S. Payette, “Spectral/Hp Finite Element Models for Fluids and Structures.,” Tech. Rep. SAND2012-7615, Sandia National Laboratories, 2012.
- [44] C. D. Cantwell, S. Yakovlev, R. M. Kirby, N. S. Peters, and S. J. Sherwin, “High-order spectral/hp element discretisation for reaction–diffusion problems on surfaces: Application to cardiac electrophysiology,” *Journal of Computational Physics*, vol. 257, Part A, pp. 813–829, 2014.
- [45] C. D. Cantwell, D. Moxey, A. Comerford, A. Bolis, G. Rocco, G. Mengaldo, D. De Grazia, S. Yakovlev, J. E. Lombard, D. Ekelschot, B. Jordi, H. Xu, Y. Mohamied, C. Eskilsson, B. Nelson, P. Vos, C. Biotto, R. M. Kirby, and S. J. Sherwin, “Nektar++: An open-source spectral/hp element framework,” *Computer Physics Communications*, vol. 192, pp. 205–219, 2015.
- [46] P. Šolín, L. Dubcova, and I. Doležal, “Adaptive hp-FEM with arbitrary-level hanging nodes for Maxwell’s equations,” *Adv. Appl. Math. Mech.*, vol. 2, no. 4, pp. 518–532, 2010.
- [47] P. Šolín and L. Korous, “Adaptive higher-order finite element methods for transient PDE problems based on embedded higher-order implicit Runge–Kutta methods,” *Journal of Computational Physics*, vol. 231, no. 4, pp. 1635–1649, 2012.
- [48] N. Zander, T. Bog, S. Kollmannsberger, D. Schillinger, and E. Rank, “Multi-level hp-adaptivity: high-order mesh adaptivity without the difficulties of constraining hanging nodes,” *Computational Mechanics*, vol. 55, no. 3, pp. 499–517, 2015.
- [49] C. D. Mote, “Global-local finite element,” *International Journal for Numerical Methods in Engineering*, vol. 3, no. 4, pp. 565–574, 1971.
- [50] O. C. Zienkiewicz and A. Craig, “Adaptive refinement, error estimates, multigrid solution and hierarchic finite element method concepts,” in *Accuracy Estimates and Adaptive Refinements in Finite Element Calculations* (I. Babuska, O. C. Zienkiewicz, J. Gago, and E. R. de Oliveira, eds.), pp. 25–55, New York: Wiley, 1986.
- [51] W. F. Mitchell, “Adaptive refinement for arbitrary finite-element spaces with hierarchical bases,” *Journal of Computational and Applied Mathematics*, vol. 36, no. 1, pp. 65–78, 1991.
- [52] T. Belytschko, J. Fish, and B. E. Engelmann, “A finite element with embedded localization zones,” *Computer Methods in Applied Mechanics and Engineering*, vol. 70, no. 1, pp. 59–89, 1988.
- [53] J. Fish and T. Belytschko, “Elements with embedded localization zones for large deformation problems,” *Computers & Structures*, vol. 30, no. 1–2, pp. 247–256, 1988.
- [54] J. Fish and T. Belytschko, “A finite element with a unidirectionally enriched strain field for localization analysis,” *Computer Methods in Applied Mechanics and Engineering*, vol. 78, no. 2, pp. 181–200, 1990.
- [55] T. Belytschko, J. Fish, and A. Bayliss, “The spectral overlay on finite elements for problems with high gradients,” *Computer Methods in Applied Mechanics and Engineering*, vol. 81, no. 1, pp. 71–89, 1990.
- [56] E. Rank, “Adaptive remeshing and h-p domain decomposition,” *Computer Methods in Applied Mechanics and Engineering*, vol. 101, no. 1–3, pp. 299–313, 1992.

- [57] J. Fish, “The s-version of the finite element method,” *Computers & Structures*, vol. 43, no. 3, pp. 539–547, 1992.
- [58] J. Fish, “Hierarchical modelling of discontinuous fields,” *Communications in Applied Numerical Methods*, vol. 8, no. 7, pp. 443–453, 1992.
- [59] J. Fish and S. Markolefas, “Adaptive s-method for linear elastostatics,” *Computer Methods in Applied Mechanics and Engineering*, vol. 104, no. 3, pp. 363–396, 1993.
- [60] J. Fish, S. Markolefas, R. Guttal, and P. Nayak, “On adaptive multilevel superposition of finite element meshes for linear elastostatics,” *Applied Numerical Mathematics*, vol. 14, no. 1–3, pp. 135–164, 1994.
- [61] P. K. Moore and J. E. Flaherty, “Adaptive local overlapping grid methods for parabolic systems in two space dimensions,” *Journal of Computational Physics*, vol. 98, no. 1, pp. 54–63, 1992.
- [62] I. Babuška and J. M. Melenk, “The Partition of Unity Method,” *International Journal for Numerical Methods in Engineering*, vol. 40, no. 4, pp. 727–758, 1997.
- [63] T. Strouboulis, I. Babuška, and K. Copps, “The design and analysis of the Generalized Finite Element Method,” *Computer Methods in Applied Mechanics and Engineering*, vol. 181, no. 1–3, pp. 43–69, 2000.
- [64] T.-P. Fries and T. Belytschko, “The extended/generalized finite element method: An overview of the method and its applications,” *International Journal for Numerical Methods in Engineering*, vol. 84, no. 3, pp. 253–304, 2010.
- [65] X.-P. Xu and A. Needleman, “Void nucleation by inclusion debonding in a crystal matrix,” *Modelling and Simulation in Materials Science and Engineering*, vol. 1, no. 2, pp. 111–132, 1993.
- [66] M. J. van den Bosch, P. J. G. Schreurs, and M. G. D. Geers, “An improved description of the exponential Xu and Needleman cohesive zone law for mixed-mode decohesion,” *Engineering Fracture Mechanics*, vol. 73, no. 9, pp. 1220–1234, 2006.
- [67] Q. Yang and B. Cox, “Cohesive models for damage evolution in laminated composites,” *International Journal of Fracture*, vol. 133, no. 2, pp. 107–137, 2005.
- [68] T. J. R. Hughes, *The finite element method: linear static and dynamic finite element analysis*. Mineola, NY: Dover Publications, 2000.
- [69] K. J. Bathe, *Finite element procedures*. New Jersey: Prentice Hall, 2007.
- [70] J. C. J. Schellekens and R. de Borst, “On the numerical integration of interface elements,” *International Journal for Numerical Methods in Engineering*, vol. 36, no. 1, pp. 43–66, 1993.
- [71] Y. Guo and M. Ruess, “Nitsche’s method for a coupling of isogeometric thin shells and blended shell structures,” *Computer Methods in Applied Mechanics and Engineering*, vol. 284, pp. 881–905, 2015.
- [72] M. Ruess, D. Schillinger, A. I. Özcan, and E. Rank, “Weak coupling for isogeometric analysis of non-matching and trimmed multi-patch geometries,” *Computer Methods in Applied Mechanics and Engineering*, vol. 269, pp. 46–71, 2014.

- [73] L. Demkowicz, J. T. Oden, W. Rachowicz, and O. Hardy, “Toward a universal h-p adaptive finite element strategy, part 1. Constrained approximation and data structure,” *Computer Methods in Applied Mechanics and Engineering*, vol. 77, no. 1–2, pp. 79–112, 1989.
- [74] M. Ainsworth and J. T. Oden, “A posteriori error estimation in finite element analysis,” *Computer Methods in Applied Mechanics and Engineering*, vol. 142, no. 1–2, pp. 1–88, 1997.
- [75] P. Frauenfelder, *hp-Finite element methods on anisotropically, locally refined meshes in three dimensions with stochastic data*. PhD thesis, Swiss Federal Institute of Technology Zürich, Zürich, 2004.
- [76] P. Šolín, *Higher-order finite element methods*. Studies in advanced mathematics, Boca Raton: Chapman & Hall/CRC, 2004.
- [77] P. Šolín, J. Červený, and I. Doležel, “Arbitrary-level hanging nodes and automatic adaptivity in the hp-FEM,” *Mathematics and Computers in Simulation*, vol. 77, no. 1, pp. 117–132, 2008.
- [78] N. Zander, T. Bog, S. Kollmannsberger, and E. Rank, “The multi-level hp-method for three-dimensional problems: dynamically changing high-order mesh refinement with arbitrary hanging nodes,” *Computer Methods in Applied Mechanics and Engineering*, vol. submitted, 2016.
- [79] B. A. Szabó and I. Babuska, *Finite element analysis*. New York: John Wiley & Sons, 1991.
- [80] B. A. Szabó, A. Düster, and E. Rank, “The p-version of the finite element method,” in *Encyclopedia of Computational mechanics* (E. Stein, ed.), Chichester, West Sussex: John Wiley & Sons, 2004.
- [81] G. Strang, *An analysis of the finite element method*. Englewood Cliffs, N.J: Prentice-Hall, 1973.
- [82] D. Schillinger, A. Düster, and E. Rank, “The hp-d-adaptive finite cell method for geometrically nonlinear problems of solid mechanics,” *International Journal for Numerical Methods in Engineering*, vol. 89, no. 9, pp. 1171–1202, 2012.
- [83] E. Rank and R. Krause, “A multiscale finite-element method,” *Computers & structures*, vol. 64, no. 1, pp. 139–144, 1997.
- [84] R. Krause and E. Rank, “Multiscale computations with a combination of the h- and p-versions of the finite-element method,” *Computer Methods in Applied Mechanics and Engineering*, vol. 192, no. 35-36, pp. 3959–3983, 2003.
- [85] A. Düster, A. Niggel, and E. Rank, “Applying the hp-d version of the FEM to locally enhance dimensionally reduced models,” *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 37-40, pp. 3524–3533, 2007.
- [86] D. Schillinger and E. Rank, “An unfitted hp-adaptive finite element method based on hierarchical B-splines for interface problems of complex geometry,” *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 47-48, pp. 3358–3380, 2011.
- [87] D. Schillinger, L. Dedè, M. A. Scott, J. A. Evans, M. J. Borden, E. Rank, and T. J. Hughes, “An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces,” *Computer Methods in Applied Mechanics and Engineering*, vol. 249-252, pp. 116–150, 2012.
- [88] D. Schillinger, *The p- and B-spline versions of the geometrically nonlinear finite cell method and hierarchical refinement strategies for adaptive isogeometric and embedded domain analysis*. PhD thesis, Technische Universität München, Chair for Computation in Engineering, 2012.

- [89] D. Schillinger, J. A. Evans, A. Reali, M. A. Scott, and T. J. R. Hughes, “Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations,” *Computer Methods in Applied Mechanics and Engineering*, vol. 267, pp. 170–232, 2013.
- [90] M. Ainsworth, *A posteriori error estimation in finite element analysis*. Pure and applied mathematics, New York: Wiley, 2000.
- [91] B. A. Szabó, “Estimation and Control of Error Based on p Convergence,” in *Accuracy Estimates and Adaptive Refinements in Finite Element Calculations* (I. Babuska, O. C. Zienkiewicz, J. Gago, and E. R. de Oliviera, eds.), pp. 25–55, New York: Wiley, 1986.
- [92] Z. Yosibash, *Singularities in elliptic boundary value problems and elasticity and their connection with failure initiation*. No. 37 in Interdisciplinary applied mathematics, New York, NY: Springer, 2012.
- [93] I. Babuška and B. Guo, “The h-p Version of the Finite Element Method for Domains with Curved Boundaries,” *SIAM Journal on Numerical Analysis*, vol. 25, no. 4, pp. 837–861, 1988.
- [94] J. R. Shewchuk, “An introduction to the conjugate gradient method without the agonizing pain,” tech. rep., Carnegie-Mellon University. Department of Computer Science, 1994.
- [95] J. M. Melenk, K. Gerdes, and C. Schwab, “Fully discrete hp-finite elements: fast quadrature,” *Computer Methods in Applied Mechanics and Engineering*, vol. 190, no. 32–33, pp. 4339–4364, 2001.
- [96] C. V. Verhoosel, J. J. C. Remmers, and M. A. Gutiérrez, “A dissipation-based arc-length method for robust simulation of brittle and ductile failure,” *International Journal for Numerical Methods in Engineering*, vol. 77, no. 9, pp. 1290–1321, 2009.
- [97] M. Bischoff, K.-U. Bletzinger, W. A. Wall, and E. Ramm, “Models and Finite Elements for Thin-Walled Structures,” in *Encyclopedia of Computational Mechanics*, Chichester, West Sussex, England: John Wiley & Sons, Ltd, 2004.
- [98] E. Rank, A. Düster, V. Nübel, K. Preusch, and O. Bruhns, “High order finite elements for shells,” *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 21-24, pp. 2494–2512, 2005.
- [99] J. Parvizian, A. Düster, and E. Rank, “Finite cell method,” *Computational Mechanics*, vol. 41, no. 1, pp. 121–133, 2007.
- [100] A. Düster, J. Parvizian, Z. Yang, and E. Rank, “The finite cell method for three-dimensional problems of solid mechanics,” *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 45–48, pp. 3768–3782, 2008.
- [101] D. Schillinger, M. Ruess, N. Zander, Y. Bazilevs, A. Düster, and E. Rank, “Small and large deformation analysis with the p- and B-spline versions of the Finite Cell Method,” *Computational Mechanics*, vol. 50, no. 4, pp. 445–478, 2012.
- [102] T. Bog, N. Zander, S. Kollmannsberger, and E. Rank, “Normal contact with high order finite elements and a fictitious contact material,” *Computers & Mathematics with Applications*, vol. 70, no. 7, pp. 1370–1390, 2015.
- [103] M. Ruess, D. Tal, N. Trabelsi, Z. Yosibash, and E. Rank, “The finite cell method for bone simulations: verification and validation,” *Biomechanics and modeling in mechanobiology*, vol. 11, no. 3-4, pp. 425–37, 2012.

- [104] C. Verhoosel, G. van Zwieten, B. van Rietbergen, and R. de Borst, “Image-based goal-oriented adaptive isogeometric analysis with application to the micro-mechanical modeling of trabecular bone,” *Computer Methods in Applied Mechanics and Engineering*, vol. 284, pp. 138–164, 2015.
- [105] E. Rank, S. Kollmannsberger, C. Sorger, and A. Düster, “Shell Finite Cell Method: A high order fictitious domain approach for thin-walled structures,” *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 45-46, pp. 3200–3209, 2011.
- [106] M. Ruess, D. Schillinger, Y. Bazilevs, V. Varduhn, and E. Rank, “Weakly enforced essential boundary conditions for NURBS-embedded and trimmed NURBS geometries on the basis of the finite cell method,” *International Journal for Numerical Methods in Engineering*, vol. 95, no. 10, pp. 811–846, 2013.
- [107] Y. Guo and M. Ruess, “A layerwise isogeometric approach for NURBS-derived laminate composite shells,” *Composite Structures*, vol. 124, pp. 300–309, 2015.
- [108] L. Kudela, N. Zander, T. Bog, S. Kollmannsberger, and E. Rank, “Efficient and accurate numerical quadrature for immersed boundary methods,” *Advanced Modeling and Simulation in Engineering Sciences*, vol. 2, no. 1, pp. 1–22, 2015.
- [109] D. Kamensky, M.-C. Hsu, D. Schillinger, J. A. Evans, A. Aggarwal, Y. Bazilevs, M. S. Sacks, and T. J. R. Hughes, “An immersogeometric variational framework for fluid–structure interaction: Application to bioprosthetic heart valves,” *Computer Methods in Applied Mechanics and Engineering*, vol. 284, pp. 1005–1053, 2015.
- [110] D. Schillinger and M. Ruess, “The Finite Cell Method: A Review in the Context of Higher-Order Structural Analysis of CAD and Image-Based Geometric Models,” *Archives of Computational Methods in Engineering*, vol. 22, no. 3, pp. 391–455, 2014.
- [111] N. Zander, T. Bog, M. Elhaddad, R. Espinoza, H. Hu, A. Joly, C. Wu, P. Zerbe, A. Düster, S. Kollmannsberger, J. Parvizian, M. Ruess, D. Schillinger, and E. Rank, “FCMLab: A finite cell research toolbox for MATLAB,” *Advances in Engineering Software*, vol. 74, pp. 49–63, 2014.