An open source program to generate zero-thickness cohesive interface elements

Vinh Phu Nguyen[1,*]

School of Engineering, Institute of Mechanics and Advanced Materials, Cardiff University, Queen's Buildings, The Parade, Cardiff CF24 3AA

Abstract

An open source program to generate zero-thickness cohesive interface elements in existing finite element discretizations is presented. This contribution fills the gap in the literature that, to the best of the author's knowledge, there is no such program exists. The program is useful in numerical modeling of material/structure failure using cohesive interface elements. The program is able to generate one/two dimensional, linear/quadratic cohesive elements (i) at all inter-element boundaries, (ii) at material interfaces and (iii) at grain boundaries in polycrystalline materials. Algorithms and utilization of the program is discussed. Several two dimensional and three dimensional fracture mechanics problems are given including debonding process of material interfaces, multiple delamination of composite structures, crack propagation in polycrystalline structures.

Keywords: interface cohesive elements, fracture mechanics, cohesive zone models, finite element method, open source code

1. Introduction

Cohesive crack models (CM) which were pioneered in [1, 2] is a continuation of linear elastic fracture mechanics with which the unrealistic stress singularity ahead the crack tip is avoided. Application of CMs as fracture models used in the context of the finite element method however appeared substantially later in [3]. From a numerical point of view, CMs have been incorporated in a finite element (FE) context using zero-thickness interface elements, elements with embedded discontinuities and elements with discontinuous enrichment via the extended/generalized finite element method (XFEM/GFEM). A comparative study on the modelling of discontinuous fracture using these techniques was given in [4] and a review of computational methods for fracture in quasi-brittle solids has been recently reported in [5]. It should be emphasized that the term "cohesive elements" usually used to refer to cohesive interface elements is misleading since elements with embedded cohesive cracks or XFEM with cohesive cracks are also cohesive elements. Therefore, in this contribution, we suggest the name "cohesive interface elements" (in subsequent discussion interface elements are used for brevity) to indicate interface elements equipped with a cohesive law.

Elements with embedded discontinuities are a powerful tool to model crack propagation, see e.g., [6, 7, 8, 9], among others. Its most distinct advantage is the simplicity of implementation into existing FE codes compared to XFEM. Recently XFEM, which is a local partition of unity (PUM) [10] based finite element method, pioneered in [11] has become the dominant numerical method for both weak discontinuities (material interfaces) [12, 13, 14] and strong

*This work was carried out when the author was a PhD candidate at the Faculty of Civil Engineering and Geosciences, Delft University of Technology, The Netherlands under the supervision of Prof. Bert Sluys.

[1]nguyenpv@cardiff.ac.uk

discontinuities (cracks) see e.g., [15, 16, 17] among others. Its popularity is arguably due to the fact that XFEM can be applied to numerous varieties of problems (fluid mechanics, biofilm growth, multiphysics etc.). However, it comes with complications including numerical integration of elements crossed by the discontinuities, complexity concerning the implementation of the method in existing finite element packages. Furthermore, finding the enrichment functions to model intersecting discontinuities is not a feasible task, especially for three dimensions. Also, parallelizing the XFEM is not straightforward.

For polycrystalline structures, a common fracture phenomenon is inter-granular cracking i.e., cracks propagate along the grain boundaries. This fracture mechanism is naturally modelled with interface elements that are inserted, prior to computation, along the grain boundaries because the crack path is known in advance. Recently, this problem was tackled with the generalized finite element method (GFEM) [18]. The advantage of GFEM is that only a structured simple mesh is required since the grain topology is independent of the mesh and modelled with special enrichment functions. However extending these enrichment functions to three dimensions and intra-granular fracture is not easy. Moreover, imposing Dirichlet boundary conditions on enriched nodes is not a trivial task. In contrary, using interface elements, both two dimensional (2D) and three dimensional (3D) inter and intra-granular fracture can be modelled. Such a 2D application has been recently reported in [19] and 3D cases in [20].

Delamination of laminated composites has been traditionally modelled using interface elements [21, 22, 23]. Recently, PUM-based finite elements has been used to model delamination with mesh independent of the delamination surface [24, 25]. However, to model 3D delaminated composites, PUM-based method must employ either a layer of 3D solid elements or solid-like shell elements which increases the computational efforts significantly. Furthermore, PUM-based method is not suited for two dissimilar structures bonded together such as skin-stringer interface. Using interface elements, an effective cohesive element for shell analysis was presented in [26].

In addition to the natural application of interface elements for problems in which crack paths are known *a priori* such as material interface debonding [27, 28], delamination, inter-granular cracking, other advantages of interface elements include (i) crack initiation does not rely on some fracture criteria, (ii) straightforward implementation, (iii) complex fracture mechanism such as crack branching, crack coalescence can be handled with ease [29, 30, 19, 31, 20]. Other applications of interface elements are mesoscopic modeling of concrete materials [32], dynamic fracture and fragmentation of solids [33, 34] and shell fracture [35, 36]. Other computational methods that are capable of dealing with complex fracture mechanics problems (with large deformation) include meshfree methods, e.g., [37, 38, 39, 40, 41, 42], and peridynamics see [43, 44] among others.

There are two ways by which one can implement interface elements. In the first approach, interface elements are introduced, where necessary, before the simulation starts. The cohesive laws used in this approach is referred to as *intrinsic cohesive laws*. In the second approach, interface elements are inserted during the simulation and the corresponding cohesive laws are referred to as *extrinsic cohesive laws*. Both approaches have their own advantages and shortcomings. In the former, the shortcomings are (i) mesh sensitivity (this however does not apply for problems in which the crack path is known in advance, for example delamination analysis, inter-granular fracture of polycrystals or debonding of material interfaces) and (ii) reliance on a high dummy stiffness to model the perfect bond prior to fracture. Mesh sensitivity issue can be eliminated with sufficiently refined random meshes. For the artificial compliance introduced by the interface elements, one remedy is to initially constraint the interface elements (using the master-slave method [19] or a discontinuous Galerkin [45, 20]) and only active them when necessary. Another possibility is to find the proper value for the dummy stiffness for interface elements as in [46]. In the latter, by only introducing interface elements when a fracture criterion is met all the shortcomings of intrinsic cohesive elements are removed. However, it brings complex implementation and parallelism issues.

Although implementation of intrinsic cohesive elements is straightforward (and thus available in major commercial FE packages), the pre-processing stage in which one has to insert interface elements into an existing finite element mesh is quite complicated and there does not exist any open source tool to do that. The aim of this manuscript is therefore to present a simple pre-processing program which is able to insert one and two dimensional interface elements in a finite

element mesh. This program is independent of mesh generation programs and freely downloadable on our website. We believe that it is helpful for researchers doing fracture analysis using interface elements, which play an important role despite of the emergence of novel methods such as XFEM, but do not have access to commercial preprocessors. It should also be useful for Discontinuous Galerkin based analyses [47]. The code also generates interface elements that are suitable for discontinuous Galerkin based methods. Moreover, through the open source code, adaptation to individual needs is much easier than using commercial preprocessors. The tool has been used in [48, 49, 50] and we hope through this manuscript it will be shared with the fracture and the discontinuous Galerkin communities. For completeness, this paper also presents implementation details of 2D interface elements in a nonlinear FE setting.

The structure of the rest of the paper is as follows. In Section 2, the governing equations of a cracked solid are given together with the semi-discrete equations discretized by finite elements. Section 3 presents the algorithms that are used to insert one and two dimensional interface elements in a finite element mesh. In the next section, the installation and usage of the presented program is discussed. Various numerical examples are given in Section 5 including material interface debonding, two dimensional crack propagation problems in homogeneous structure and in polycrystalline structure; two and three dimensional delamination analysis. Finally some conclusions are drawn in Section 6.

2. Finite element formulation

2.1. Strong and weak forms

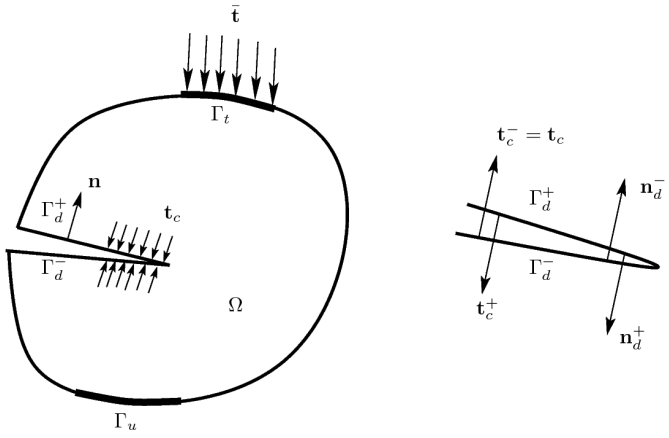


Figure 1: A two dimension solid containing a cohesive crack

The governing equations include the equilibrium equation, the natural, essential boundary conditions and the traction continuity on the crack surface

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = 0 \quad \mathbf{x} \in \Omega \tag{1a}$$

$$\mathbf{n} \cdot \boldsymbol{\sigma} = \bar{\mathbf{t}} \quad \mathbf{x} \in \Gamma_t \tag{1b}$$

$$\mathbf{u} = \bar{\mathbf{u}} \quad \mathbf{x} \in \Gamma_u \tag{1c}$$

$$\mathbf{n}_d^+ \cdot \boldsymbol{\sigma} = \mathbf{t}_c^+; \quad \mathbf{n}_d^- \cdot \boldsymbol{\sigma} = \mathbf{t}_c^-; \quad \mathbf{t}_c^+ = -\mathbf{t}_c = -\mathbf{t}_c^- \quad \mathbf{x} \in \Gamma_d \tag{1d}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor, $\mathbf{u}$ is the displacement field and $\mathbf{b}$ is the body force vector. The traction applied over the boundary Γ_t with outward unit normal vector $\mathbf{n}$ is denoted by $\mathbf{t}$, $\bar{\mathbf{u}}$ is the applied displacement over the

Dirichlet boundary Γ_u; $\Gamma_u \cup \Gamma_t = \Gamma$, $\Gamma_u \cap \Gamma_t = \emptyset$; $\mathbf{t}^{\mathrm{c}}$ is the cohesive traction across the crack Γ_d with unit normal vector $\mathbf{n}_d$. For simplicity, only small strain assumption is considered where the strain is taken as the symmetric part of the displacement gradient $\boldsymbol{\epsilon}^s = \frac{1}{2}(\nabla \mathbf{u}^s + \nabla^{\mathrm{T}} \mathbf{u}^s)$. Constitutive relations for the bulk and the cohesive crack are given later.

The weak formulation reads, see [51] for details

$$\delta W^{\mathrm{ext}} = \delta W^{\mathrm{int}} + \delta W^{\mathrm{coh}} \tag{2}$$

with

$$\delta W^{\mathrm{int}} = \int_{\Omega} \nabla^s \delta \mathbf{u} : \boldsymbol{\sigma} \mathrm{d}\Omega \tag{3a}$$

$$\delta W^{\mathrm{ext}} = \int_{\Omega} \delta \mathbf{u} \cdot \mathbf{b} \mathrm{d}\Omega + \int_{\Gamma_t} \delta \mathbf{u} \cdot \bar{\mathbf{t}} \mathrm{d}\Gamma_t \tag{3b}$$

$$\delta W^{\mathrm{coh}} = \int_{\Gamma_d} \delta [\![\mathbf{u}]\!] \cdot \mathbf{t}^{\mathrm{c}} \mathrm{d}\Gamma_d \tag{3c}$$

where $[\![\mathbf{u}]\!]$ denotes the displacement jump.

2.2. Discretization

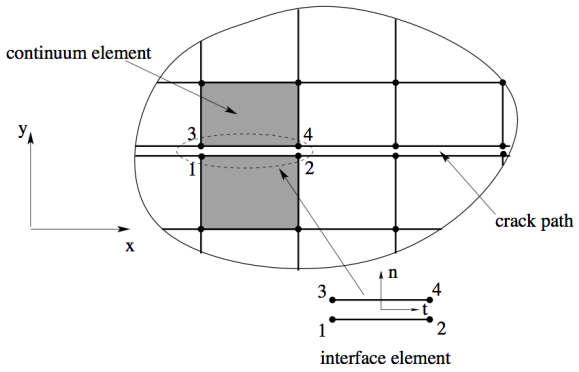


Figure 2: Discretization of the solid into continuum elements and zero-thickness interface elements.

The bulk is discretized by standard continuum elements and the discontinuity surfaces (cracks, material interfaces) are discretized by zero-thickness interface elements which are one dimensional less than the continuum elements. Fig. 2 illustrates the idea for two dimensions.

The displacement fields of the upper and lower faces of the interface element are given by

$$\mathbf{u}^+ = \mathbf{N}^{\mathrm{int}} \mathbf{u}^+, \quad \mathbf{u}^- = \mathbf{N}^{\mathrm{int}} \mathbf{u}^- \tag{4}$$

with $\mathbf{N}^{\text{int}}$ denotes the matrix of shape functions of the interface element [2] and $\mathbf{u}^+$ and $\mathbf{u}^-$ denote the nodal displacements of the upper face and lower face, respectively.

Having defined the displacement of the upper and lower faces of the interface, it is able to compute the displacement jump as

$$[\![\mathbf{u}(\mathbf{x})]\!] = \mathbf{u}^+ - \mathbf{u}^- = \mathbf{N}^{\text{int}}(\mathbf{u}^+ - \mathbf{u}^-) \tag{5}$$

where we have used Eq. (4).

For completeness, the displacement of the continuum elements and the virtual displacement are recalled here

$$\mathbf{u} = N_I \mathbf{u}_I, \quad \delta\mathbf{u} = N_I \delta\mathbf{u}_I \tag{6}$$

where N_I is the shape functions of the continuum elements.

Introducing Eqs. (6) and (5) into the weak form Eq. (2), we get the following semi-discrete equations

$$\mathbf{f}^{\text{ext}} = \mathbf{f}^{\text{int}} + \mathbf{f}^{\text{coh}} \tag{7}$$

where $\mathbf{f}^{\text{ext}}$, $\mathbf{f}^{\text{int}}$ and $\mathbf{f}^{\text{coh}}$ are the external force, internal force and cohesive force vectors, respectively.

The external and internal force vectors are computed from contributions of continuum elements and given by

$$\mathbf{f}_e^{\text{int}} = \int_{\Omega_e} \mathbf{B}^{\mathrm{T}} \boldsymbol{\sigma} \mathrm{d}\Omega_e \tag{8}$$

$$\mathbf{f}_e^{\text{ext}} = \int_{\Omega_e} \mathbf{N}^{\mathrm{T}} \mathbf{b} \mathrm{d}\Omega_e + \int_{\Gamma_t^e} \mathbf{N}^{\mathrm{T}} \bar{\mathbf{t}} \mathrm{d}\Gamma_t^e \tag{9}$$

where the shape function matrix and the strain-displacement matrix, of which expressions are standard and thus not given here, are denoted by $\mathbf{N}$ and $\mathbf{B}$.

The cohesive force vector is computed by assembling the contribution of all interface elements. It is given by for an interface element ie

$$\begin{aligned} \mathbf{f}_{ie,+}^{\text{coh}} &= \int_{\Gamma_d} (\mathbf{N}^{\text{int}})^{\mathrm{T}} \mathbf{t}^{\mathrm{c}} \mathrm{d}\Gamma \\ \mathbf{f}_{ie,-}^{\text{coh}} &= -\int_{\Gamma_d} (\mathbf{N}^{\text{int}})^{\mathrm{T}} \mathbf{t}^{\mathrm{c}} \mathrm{d}\Gamma \end{aligned} \tag{10}$$

where the first quantity is assembled to the total internal force vector according to the degrees of freedom (dofs) of nodes of the upper surface and the second quantity contributes to the internal force vector to locations defined by nodes of the lower surface.

[2] For 4-node element, we have

$$\mathbf{N}^{\text{int}} = \begin{bmatrix} N_1 & 0 & N_2 & 0 \\ 0 & N_1 & 0 & N_2 \end{bmatrix}$$

where N_1, N_2 are the two-node line element shape functions.

2.3. Linearisation

For the sake of simplicity, we assume that the nonlinear process is concentrated on the cohesive cracks. The bulk is thus simply a linear elastic material. The behaviour of cracks is modelled with a cohesive law or a traction separation law (TSL). The constitutive equations are then given in rate form as

$$\begin{aligned} \dot{\boldsymbol{\sigma}} &= \mathbf{D}\dot{\boldsymbol{\epsilon}} \\ \dot{\mathbf{t}}^{\mathrm{c}} &= \mathbf{T}[\![\dot{\mathbf{u}}]\!] \end{aligned} \tag{11}$$

where $\mathbf{D}$ is the bulk tangent matrix which is simply the elasticity matrix and $\mathbf{T}$ is the cohesive (discrete) tangent matrix whose precise form depends on the TSL used.

The linearization of the internal force vector Eq. (8) leads to the standard material tangent stiffness matrix which is given by

$$\mathbf{K}_e^{\mathrm{con}} = \int_{\Omega} \mathbf{B}_e^{\mathrm{T}} \mathbf{D} \mathbf{B}_e \mathrm{d}\Omega \tag{12}$$

which is regiven here only for completeness. The implementation of interface elements is independent of the bulk solid element formulation.

Substituting the displacement jump in Eq. (5) into Eq. (11) leads to

$$\dot{\mathbf{t}}^{\mathrm{c}} = \mathbf{T}\mathbf{N}^{\mathrm{int}}(\dot{\mathbf{u}}^{+} - \dot{\mathbf{u}}^{-}) \tag{13}$$

This is then transformed to the global coordinate system by using the orthogonal transformation matrix $\mathbf{Q}$

$$\dot{\mathbf{t}}^{\mathrm{c}} = \mathbf{Q}\mathbf{T}\mathbf{Q}^{\mathrm{T}}\mathbf{N}^{\mathrm{int}}(\dot{\mathbf{u}}^{+} - \dot{\mathbf{u}}^{-}) \tag{14}$$

with $\mathbf{Q}$ is given by

$$\mathbf{Q} = \begin{bmatrix} \mathbf{n} & \mathbf{s} & \mathbf{t} \end{bmatrix} \tag{15}$$

where $\mathbf{n}$ is the unit normal vector of the interface element and $\mathbf{s}$, $\mathbf{t}$ are the unit tangential vectors of the interface element. Refer to [52] for the construction of $\mathbf{Q}$ for 2D interface elements.

The linearization of the cohesive force vector Eq. (10) requires a bit of elaboration as shown

$$\begin{bmatrix} \dfrac{\partial \mathbf{f}_{ie,+}^{\mathrm{coh}}}{\partial \mathbf{u}} \\ \dfrac{\partial \mathbf{f}_{ie,-}^{\mathrm{coh}}}{\partial \mathbf{u}} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial \mathbf{f}_{ie,+}^{\mathrm{coh}}}{\partial \mathbf{u}^{+}} & \dfrac{\partial \mathbf{f}_{ie,+}^{\mathrm{coh}}}{\partial \mathbf{u}^{-}} \\ \dfrac{\partial \mathbf{f}_{ie,-}^{\mathrm{coh}}}{\partial \mathbf{u}^{+}} & \dfrac{\partial \mathbf{f}_{ie,-}^{\mathrm{coh}}}{\partial \mathbf{u}^{-}} \end{bmatrix} \begin{bmatrix} \delta\mathbf{u}^{+} \\ \delta\mathbf{u}^{-} \end{bmatrix} \tag{16}$$

Therefore, the cohesive tangent stiffness matrix for a given interface element is given by

$$\mathbf{K}_e^{\mathrm{int}} = \begin{bmatrix} \displaystyle\int_{\Gamma_d} \mathbf{N}^{\mathrm{T}}\mathbf{Q}\mathbf{T}\mathbf{Q}^{\mathrm{T}}\mathbf{N}\mathrm{d}\Gamma & -\displaystyle\int_{\Gamma_d} \mathbf{N}^{\mathrm{T}}\mathbf{Q}\mathbf{T}\mathbf{Q}^{\mathrm{T}}\mathbf{N}\mathrm{d}\Gamma \\ -\displaystyle\int_{\Gamma_d} \mathbf{N}^{\mathrm{T}}\mathbf{Q}\mathbf{T}\mathbf{Q}^{\mathrm{T}}\mathbf{N}\mathrm{d}\Gamma & \displaystyle\int_{\Gamma_d} \mathbf{N}^{\mathrm{T}}\mathbf{Q}\mathbf{T}\mathbf{Q}^{\mathrm{T}}\mathbf{N}\mathrm{d}\Gamma \end{bmatrix} \tag{17}$$

In fact only one term, which is a $2n \times 2n$ matrix with n being the number of nodes of one surface of the interface element, in the above needs to be computed, and it is then assembled to the appropriate locations. Also noting that

in the above we have omitted the superscript *int* for clarity. It is interesting to note that the above equation for the interface stiffness has the same form with the one established with the phantom node method [53, 54] or the method used by [55]. For completeness, the peusedo code of a general interface element program is given in Box 1 in which we have omitted the contribution of continuum elements since it is standard.

Box 1 Flowchart of a general interface element model

1. For a given load step and for an interface element e, do
2. Get the connectivity $inodes$
3. Get dofs of upper and lower faces, $idofsA$, $idofsB$
4. Get nodal displacements of upper and lower faces, $\mathbf{u}_A$, $\mathbf{u}_B$
5. Compute the global jump: $[\![\mathbf{u}]\!] = \mathbf{u}_A - \mathbf{u}_B$
6. Loop over integration points, ip with weight wip *
   (a) Compute displacement jump at ip: $[\![\mathbf{u}]\!]_{ip} = \mathbf{N}[\![\mathbf{u}]\!]$
   (b) Convert to local system: $[\![\mathbf{u}]\!]^{\text{loc}}_{ip} = \mathbf{Q}^{\text{T}}[\![\mathbf{u}]\!]_{ip}$
   (c) Compute local traction and tangent $\mathbf{t}_{\text{loc}}$, $\mathbf{T}$ using $[\![\mathbf{u}]\!]^{\text{loc}}_{ip}$ and a TSL
   (d) Compute the cohesive internal force vector: $\mathbf{f}^{\text{coh}} = \mathbf{f}^{\text{coh}} + \mathbf{N}^{\text{T}}\mathbf{Q}\mathbf{t}_{\text{loc}}w_{ip}$
   (e) Compute the tangent stiffness: $\mathbf{K}^{\text{coh}} = \mathbf{K}^{\text{coh}} + \mathbf{N}^{\text{T}}\mathbf{Q}\mathbf{T}\mathbf{Q}^{\text{T}}\mathbf{N}w_{ip}$ **
7. Assembling to global force and tangent ***
   (a) $\mathbf{f}[idofsA] = \mathbf{f}[idofsA] + \mathbf{f}^{\text{coh}}$
   (b) $\mathbf{f}[idofsB] = \mathbf{f}[idofsB] - \mathbf{f}^{\text{coh}}$
   (c) $\mathbf{K}[idofsA, idofsA] = \mathbf{K}[idofsA, idofsA] + \mathbf{K}^{\text{coh}}$
   (d) $\mathbf{K}[idofsB, idofsB] = \mathbf{K}[idofsB, idofsB] + \mathbf{K}^{\text{coh}}$
   (e) $\mathbf{K}[idofsA, idofsB] = \mathbf{K}[idofsA, idofsB] - \mathbf{K}^{\text{coh}}$
   (f) $\mathbf{K}[idofsB, idofsA] = \mathbf{K}[idofsB, idofsA] - \mathbf{K}^{\text{coh}}$

* Any kind of numerical integration rules can be used here. However, it has been reported that, [56] the Newton-Cotes scheme overcomes the traction oscillation issues. The integration surface Γ is generally the midsurface of the cohesive element.
** In literature, was made use of $\mathbf{K}^{\text{coh}} = \int_\Gamma \mathbf{N}^{\text{T}}\mathbf{Q}^{\text{T}}\mathbf{T}\mathbf{Q}\mathbf{N}d\Gamma$, in this case, their transformation matrix is the transpose of our.
*** Extension to large displacement case can be made by two additions. First the midsurface Γ is the one in the current deformed configuration. Second, a geometry tangent stiffness is added to $\mathbf{K}^{\text{coh}}$.

3. Automatic generation of interface elements

The implementation of interface elements into an existing finite element code is quite straightforward provided that meshes including both continuum and interface elements is available. However, creating such meshes has not been discussed in the literature yet, at least to the authors' knowledge. In this section, a simple pre-processing program [3] is presented that reads a FE mesh and modifies that mesh so that 1D and 2D interface elements can be inserted along

[3] Freely downloadable at `https://sites.google.com/site/phuvinhnguyensite/home/programs`. The code consists of about 2000 lines of C++ code and uses extensively the STL and Boost libraries.

either material interfaces or grain boundaries of a polycrystalline solid or along a surface where the crack is assumed to grow. Supported interface elements are given in Fig. 3. Note that high order B-spline interface elements given in [49] are also implemented. The program starts by reading a FE mesh including nodal coordinates, element connectivity arrays and element groups. It then builds the support for all nodes (support of a node is the set of elements sharing this node) and the neighbors of all elements. After that, it adapts the mesh by duplicating nodes and changing connectivities. Finally it generates a set of interface elements. In the following, algorithms used to insert interface elements along material interfaces and grain boundaries in 2D and 3D are going to be given in detail. We note that in an isogeometric analysis framework, interface elements can be created straightforwardly as presented in [57].

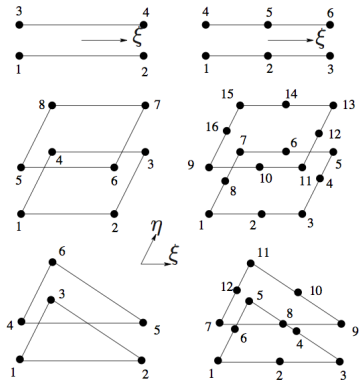


Figure 3: Supported interface elements: left column linear elements and right column quadratic elements; 1D interface elements (top row) and 2D interface elements.

3.1. Interface elements along a material interface

Fig. 4 illustrates the algorithm, adopted to place 1D cohesive elements along a material interface, which is given in detail in Box 2. In words, nodes on the material interface are cloned and elements containing this interface are divided into two groups. The first group contains elements below the interface: their connectivities are kept unchanged. The second group consists of elements above the interface, their connectivities are modified using the duplicated nodes.

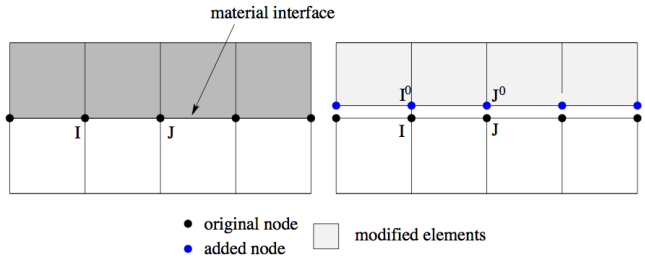


Figure 4: Insertion of interface elements along a bimaterial interface. Note that this horizontal material interface is only for illustration. The code can generate interface elements along curved material interfaces.

Box 2 Algorithm to insert interface elements along material interfaces

1. Initialization
   (a) Building nodal support
2. Detecting interface nodes
   (a) For node I, loop over its support
   (b) Count the number of materials present in this support, $nmat$
   (c) If $nmat >= 2$, then I is an interface node
3. Duplicating nodes
   (a) $i = nodeCount$, $nodeCount$ is the total number of original nodes
   (b) For an interface node I, do
      i. Clone node I: build a new node with coordinate of I, index of i
      ii. Store index of node I in $duplicatedNodes[I][0]$
      iii. Store index of this new node in $duplicatedNodes[I][1]$
      iv. $i = i + 1$
4. Tearing elements (modifying connectivities)
   (a) For node I of interface nodes, loop over support of node I, s
      i. First element in s kept unchanged, let mat be the material of this element
      ii. For element e in the rest of s,
      iii. if material of e coincides with mat, continue with next element
      iv. change node I in the connectivity of e by duplicatedNodes[I][1] *
   (b) End loop over s
5. Inserting interface elements
   (a) For element e, loop over edges s of e **
      i. If s is on external boundary or s already treated, continue next edge
      ii. Let the nodes of edge s be I and J
      iii. If either I or J is not an interfacial node, continue next edge
      iv. Add one interface element with connectivity as

$$[\ I \quad J \quad duplicatedNodes[I][1] \quad duplicatedNodes[J][1]\ ]$$

   (b) End loop over edges

* In fact, we store the original connectivity of one element in an array $inodes0$ and the new (modified) one in $inodes$.
** For quadratic elements, the edge s connects the two corner nodes I and J and the midside node of s is P. Then the interface element's connectivity is

$$[\ I \quad P \quad J \quad duplicatedNodes[I][1] \quad duplicatedNodes[P][1] \quad duplicatedNodes[J][1]\ ]$$

Noting that this is dependent on the node numbering convention of the FE code.

To generate 2D cohesive surfaces along 2D material interfaces, the procedure in Box 2 needs only be modified in step 5 by looping over the faces rather than the edges. To this end, after reading the mesh, the program constructs the faces of all elements. For quadratic 3D solid elements, two kinds of face are created, one consists of corner nodes of that face and the other contains all nodes. The former is used to check whether a given face f is interfacial i.e., lying on a material interface and f is treated already. The later is used to build the connectivity of the inserted interface elements.

3.2. Interface elements at grain boundaries

The algorithm in Box 2 has been extended with minor modifications to insert cohesive elements along grain boundaries of polycrystalline solids. To this end, elements belong to different grains are assigned to different materials (more precisely different groups) when meshing the domain as shown in Fig. 5(a). Thus, the grain boundaries can be considered as material interfaces. The procedure is given in Box 3 with the only difference from Box 2 lies in the treatment of junction nodes (e.g., those have three materials present in their supports) because for a junction node there are two duplicated nodes whereas for a normal interfacial node there is only one duplicated node. To tear elements around a junction node I, refer to Fig. 5(b), elements belonging to one grain are kept unchanged, elements in the second grain will be modified using the first duplicated node of I and finally elements in the third grain will be changed to the second duplicated node of I. This is exactly what is given in step 4 of Box 3. To insert an interface element along the edge s, connecting nodes I and J, of element ie where s connects one junction node or two, we find the neighbor element je sharing the same edge s. Knowing the positions of I and J in the original connectivity $jnodes0$ of je, it is able to find the indices of the duplicated nodes of I and J by using the new connectivity $jnodes$ of je. See the Fig. 5(c) for an illustration. It is noting that in some special cases, there are edges (KJ in Fig. 5(d)) connecting interfacial nodes but they are not on grain boundaries. Thus, no interface elements are created there. The extension to 3D polycrystals is straightforward and also implemented in the program.

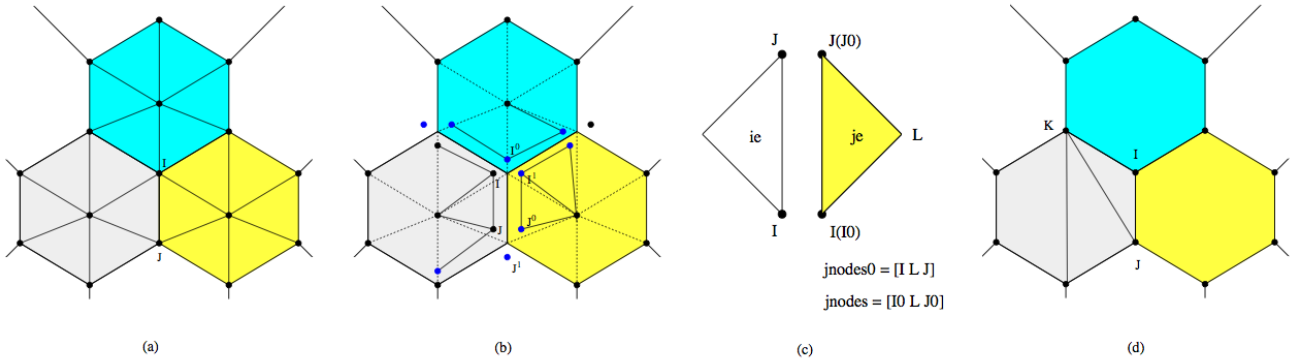


Figure 5: Insertion of interface elements along grain boundaries of polycrystal: (a) original FE mesh, (b) modified mesh, (c) inserted interface and (d) special case (KJ connecting interfacial nodes but it is not on grain boundaries. Thus, no interface element is created there).

3.3. Interface elements at every inter-element boundaries

The program is also capable of inserting cohesive elements in between every inter-element boundaries. This is useful for discontinuous Galerkin problems and for dynamic fracture and fragmentation problems. In this case, at a node I,

Box 3 Algorithm to insert interface elements along grain boundaries

1. Initialization
   (a) Building nodal support
   (b) Building neighbors for all elements
2. Detecting interface nodes (see Box 2)
3. Duplicating nodes
4. Tearing elements (modifying connectivities)
   (a) For node I of interface nodes, do
   (b) If I is not a junction node, following Box 2
   (c) Get support of node I, s
   (d) $j = 0$
   (e) For element ie in s, do
      i. Let material of ie be $imat$
      ii. For element je in s, do
         A. Ignore elements treated ($je < ie$)
         B. Let material of je be $jmat$
         C. If $jmat \neq imat$ continue with next element
         D. Change connectivity of je at node I by $duplicatedNodes[I][j]$
      iii. End loop over je
      iv. $j = j + 1$
   (f) End loop over ie
5. Inserting interface elements
   (a) For element ie, loop over edges of ie, s
      i. If s is on external boundary or s already treated, continue next edge
      ii. Let the nodes of edge s be I and J,
      iii. If neither of I nor J is a junction node, following Box 2
      iv. Loop over neighbors of ie, then find element je sharing the edge s with ie
      v. Get original connectivity of je, say $jnodes0$
      vi. Get new connectivity of je, say $jnodes$
      vii. Denote positions of I and J in $jnodes0$ as $p1$, $p2$
      viii. Add a cohesive element with connectivity

$$[\ I \quad J \quad jnodes[p1] \quad jnodes[p2]\ ]$$

   (b) End loop over s

there are ($duplicity - 1$) duplicated nodes where *duplicity* is the size of the support of I, Fig. 6. To tear elements in the support s of a node I, the first element in s uses the first duplicated node of I (which is essentially node I), the second uses the second duplicated node of I and so on.

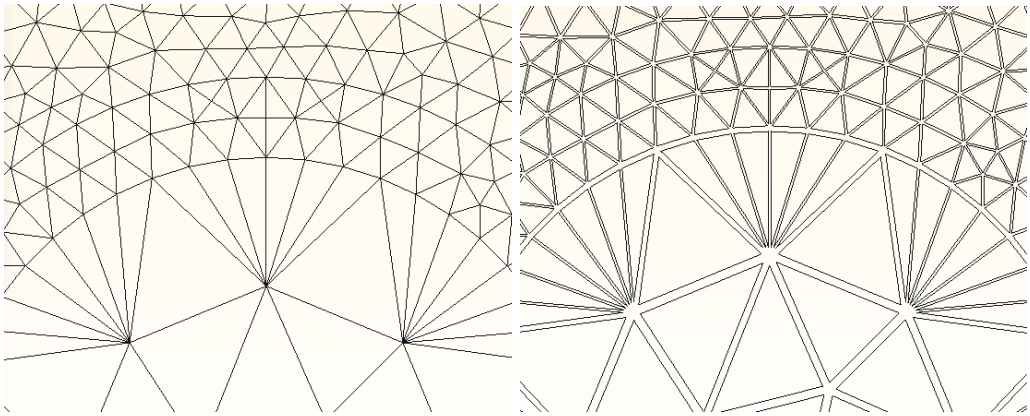


Figure 6: Insertion of interface elements at every inter-element boundaries.

Extension to the case where interface elements are not allowed at a certain domain (for example for hard inclusions in a composite material) is straightforward as demonstrated in Fig. 7. There are two differences: interfacial nodes and nodes locate in the domain where interface elements are not allowed (restricted domain). To tear elements in the support s of an interfacial node I, all the elements in s that belong to the restricted domain keep their connectivities, other elements use the duplicated nodes of node I to change their connectivities (at location of node I). Nodes within the restricted domain are skipped in the loop over all nodes of the mesh.

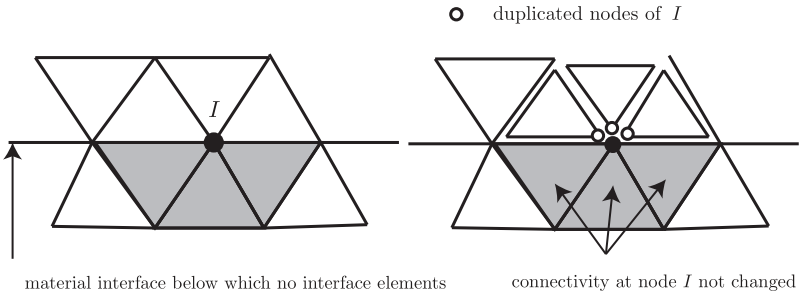


Figure 7: Insertion of interface elements at every inter-element boundaries except for a group of elements (grey elements).

3.4. Discontinuous Galerkin interface elements

In the formulation of the cohesive interface elements one has to compute the displacement jump which is essentially the difference between the displacement of the upper and lower faces. Therefore, evaluating the jump needs only the displacement of the element edges locating on either side of the interface. The volumetric elements are not needed for the

computation of the cohesive terms. The situation, however, changes for discontinuous Galerkin (dG) methods, see e.g., [47, 34]. In dG methods, the FE displacement field is discontinuous across interelement boundaries (this is achieved by inserting interface elements along every interelement boundaries) and continuity in the displacement is weakly enforced by introducing extra terms which involve, in addition to $[\![\mathbf{u}]\!]$, also the averaged stress field $\boldsymbol{\sigma} = 0.5(\boldsymbol{\sigma}^{+} + \boldsymbol{\sigma}^{-})$; where $\boldsymbol{\sigma}^{+/-}$ denote the stresses at either side of an interface. It has been shown that [34] dG methods can be implemented using interface elements. However, in order to compute the averaged stresses, the two volumetric elements to which the interface element is attached, see Fig. 8, are needed. In order to support dG methods, or the hybrid dG/cohesive interface elements [45, 34, 20], the previously presented algorithms were updated (with a slight modification) to generate the indices of volumetric elements that attach to the interface elements. Note that this is also useful for analyses based on triaxiality cohesive elements [58] or dG based rotation-free shell elements [36].

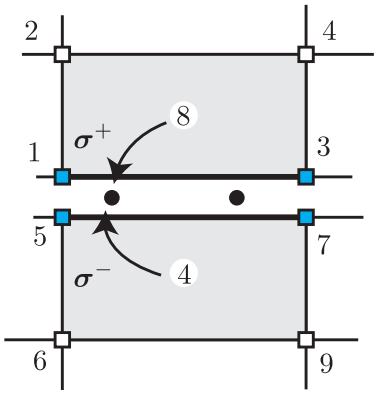


Figure 8: Discontinuous Galerkin interface elements: in addition to nodes at the interface (nodes 1,3,5,7) the two volumetric elements this interface links to (elements 4,8) are also needed in evaluating the interface terms.

4. Installation and usage

The program is entirely written in object-oriented C++ and uses the Standard Template Library (STL) and the Boost library which can be downloaded at `www.boost.org`. The program can be compiled using the scons program, (`www.scons.org`) or using Gnu Make. It has been compiled and tested on Ubuntu and Mac OS machines. By issuing the command *./interface-elem –help* on a terminal, one gets the help given in List 1.

Listing 1: Simple interface of the interface generator.

```
USAGE:
   * --mesh-file       FILE          set the file containing the mesh
   * --out-file        FILE          set the file containing the modified mesh
   * --interface-file  FILE          set the file containing the interface mesh
   * --paraview-file   FILE          set the file of ParaView format
   * --interface                     generate interface elements along material interface
   * --everywhere                    generate interface elements at all interelement boundaries
   * --domain          domNum        not generate interface elements in domain number domNum
   * --polycrystal                   generate interface elements along intergranular boundaries
   * --notch           x1 y1 x2 y2   existing notch(duplicate nodes but no interface there)
   * --notches         x1 y1 x2 y2 x3 y3 ... existing notches(duplicate nodes but no interface there)
   * --noInterface     x1 y1 x2 y2   no duplicated nodes, no interface elements along this line
```

```
13    * --isContinuum     1 or 0        continuum interface elements or discrete elements
14    * --help                          print this help and exit
```

Currently the code only supports Gmsh meshes [59] and the FE solver jem-jive [60]. In other words, it reads a Gmsh mesh file, generate the interface elements and write output files in jem-jive format. However extension can be made to support other formats. An Abaqus extension is about to finish. As can be seen, the program generates two output files–one stores the modified mesh of bulk elements and the second contains the mesh of interface elements. The option "–isContinuum" is used to generate either standard cohesive interface elements or discrete spring elements [61]. The option "–domain domNum" is used to not create interface elements in a domain numbered domNum. This is useful for instance if one does not allow hard inclusions (embedded in a soft matrix) to be cracked.

5. Numerical examples

In this section some numerical results will be presented to illustrate the utility of the proposed meshing program and the efficiency of the energy-based arc-length control [62, 63, 64] in modelling material failure. The quasi-static examples include debonding of a single fibre in an epoxy matrix, multiple delamination of a composite beam and inter-granular cracking of a notched polycrystalline sample. A dynamic fracture example using the hybrid discontinuous Galerkin/cohesive elements is also given. The original meshes (one without interface elements) have been generated using Gmsh [59].

5.1. Debonding of material interface

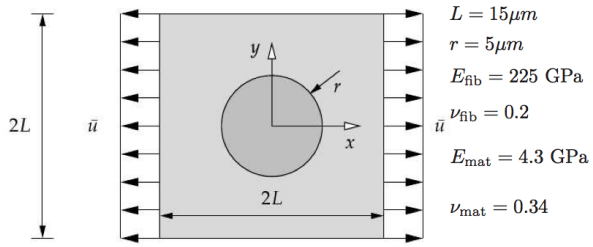


Figure 9: Geometry and loading conditions of a single fibre in an epoxy matrix.

Considering a single fibre in an epoxy matrix as shown in Fig. 9 [65]. The specimen is subjected to prescribed displacement in the x direction while the top and bottom edges are constrained in the y direction. The bulk materials are taken as linear elastic materials with parameters given in Fig. 9 whereas the interface is modelled by the Xu-Needleman's TSL [29] with $\sigma_{\max} = 50$ MPa, $q = 1$, $r = 0$ and $\delta_n = \delta_t = 10^{-5}$ mm. The weighting factor β used to compute the effective displacement jump is taken as 2.3 [66]. Due to symmetry, only a quarter of the specimen is modelled and the plane strain condition is assumed.

Given the displacement jump (Δ_n, Δ_t), the traction according to the Xu-Needleman's TSL [29] is written as

$$
\begin{aligned}
t_n &= -\frac{\phi_n}{\delta_n}\exp\left(-\frac{\Delta_n}{\delta_n}\right)\left\{\left(-r+\frac{\Delta_n}{\delta_n}\right)\frac{1-q}{r-1} - \left[q+\frac{r-q}{r-1}\left(\frac{\Delta_n}{\delta_n}-1\right)\right]\exp\left(-\frac{\Delta_t^2}{\delta_t^2}\right)\right\} \\
t_t &= 2\frac{\phi_n}{\delta_t^2}\Delta_t\left[q+\left(\frac{r-q}{r-1}\right)\frac{\Delta_n}{\delta_n}\right]\exp\left(-\frac{\Delta_n}{\delta_n}\right)\exp\left(-\frac{\Delta_t^2}{\delta_t^2}\right)
\end{aligned}
\tag{18}
$$

where δ_n and δ_t are the characteristic separations in the sense that $t_n(\delta_n) = \sigma_{\max}$ and $t_t(\delta_t/\sqrt{2}) = \tau_{\max}$ with $\sigma_{\max}$ being the maximum value of the normal traction (also called tensile strength) and $\tau_{\max}$ denotes the ultimate shear traction. Furthermore, the mode-mixity of the model is controlled by the parameters q and r. The first parameter denotes the ratio of the fracture energy in the normal and shear direction $q = \phi_t/\phi_n$, the second parameter controls the magnitude of the normal opening Δ_n^* in case of a complete pure shear separation when the normal traction is zero: $r = \Delta_n^*/\delta_n$. In Eq. (18), ϕ_n and ϕ_t represent the amount of work needed for complete separation. Using the definitions of the characteristic separations $t_n(\delta_n) = \sigma_{\max}$ and $t_t(\delta_t/\sqrt{2}) = \tau_{\max}$, one can determine ϕ_n and ϕ_t as follows

$$\phi_n = \sigma_{\max} \exp(1)\delta_n, \quad \phi_t = \tau_{\max}\sqrt{\exp(1)/2}\delta_t, \tag{19}$$

An unstructured mesh consists of 3440 quadratic triangular elements and 31 six-node interface elements is adopted. Concerning the solver, the simulation starts with load control of constant load step equals 0.1 N up to step 6 where the released energy G exceeded $5 \cdot 10^{-8}$ Nmm the solver is switched to the arc-length control. The load versus prescribed displacement curve is shown in Fig. 10 together with the deformed configuration. It can be shown that the energy arc-length control is able to capture both the limit point and the snapback behavior in a low number of load steps. The number of iterations of all steps around this critical point is only 3 which proves the robustness of the method.

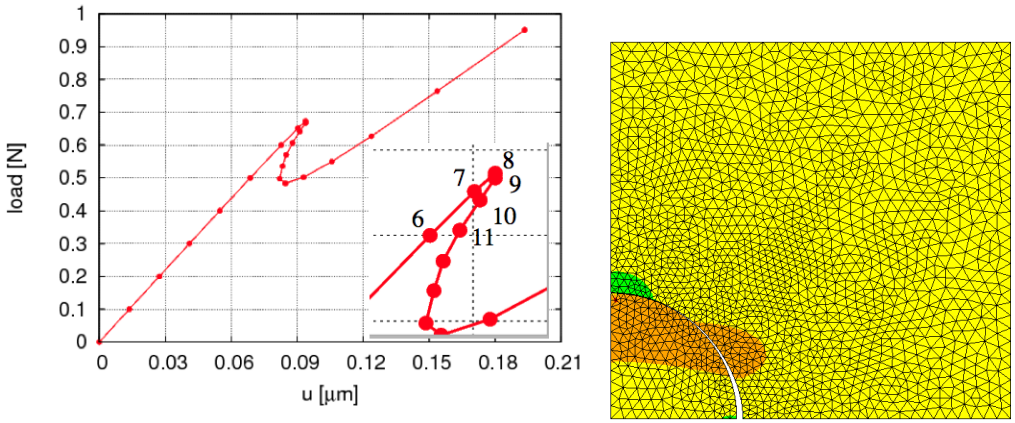


Figure 10: Load versus prescribed displacement of a single fibre in an epoxy matrix and deformed configuration.

5.2. Multi-delamination of a double cantilever beam

This example addresses a multiple mixed-mode delamination analysis of a composite specimen. The geometry and loading is given in Fig. 11. This problem was studied several times see e.g., [23, 67]. There are two initial cracksthe first initial crack on the left-hand of the specimen is placed along the mid plane and the second initial crack is positioned 20 mm on the right of and two plies below the first crack. In the numerical model, a plane strain state is assumed and material properties, which are taken from [23], is given in Table 1. The interface stiffness is $k = 105$ N/ mm^3. In this example, we adopt the damage-based bilinear cohesive law developed in [68, 69].

Since there are two potential delamination surfaces, one is the interface between layers 12 and 13, the second one is the interface between layers 10 and 11, the mesh has been built in such a way that elements on either sides of those delamination surfaces are assigned to different groups (indicated by different colors in Fig. 12). Following the standard procedure given in Box 2 would generate interface elements along the whole delamination surfaces which is

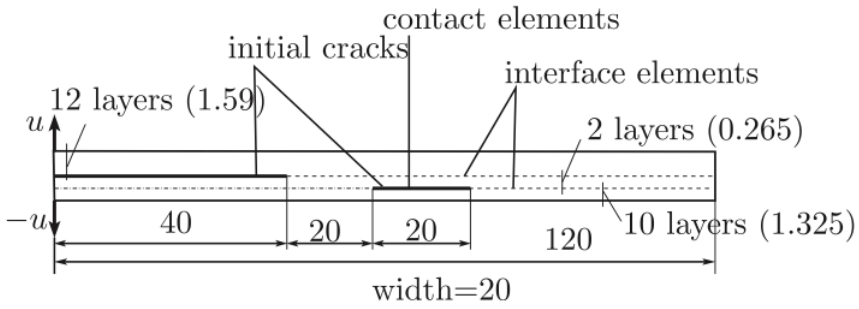


Figure 11: Multiple delamination analysis: geometry and loading.

Figure 12: Multiple delamination analysis: FE discretization showing different element groups.

E_{11}	E_{22}	G_{12}	$\nu_{12} = \nu_{13} = \nu_{23}$	
115 GPa	8.5 GPa	4.5 GPa	0.29	
G_{Ic}	G_{IIc}	τ_1^0	τ_3^0	μ
0.33 N/mm	0.8 N/mm	7.0 MPa	3.3 MPa	2.0

Table 1: Multiple delamination analysis: material properties.

obviously not desired. So, the program has been implemented an option not to add interface elements along existing notches (however, the nodes on those notches are still duplicated). Furthermore, for nodes fall within $0 \leq x \leq 60$ and $y = -0.265$, albeit interfacial nodes, they are not duplicated, thus no interface elements are created there. A mesh of 5×360 four-node quadrilateral elements and 361 four-node interface elements is adopted. Contact elements are placed along the second initial crack to avoid interpenetration there.

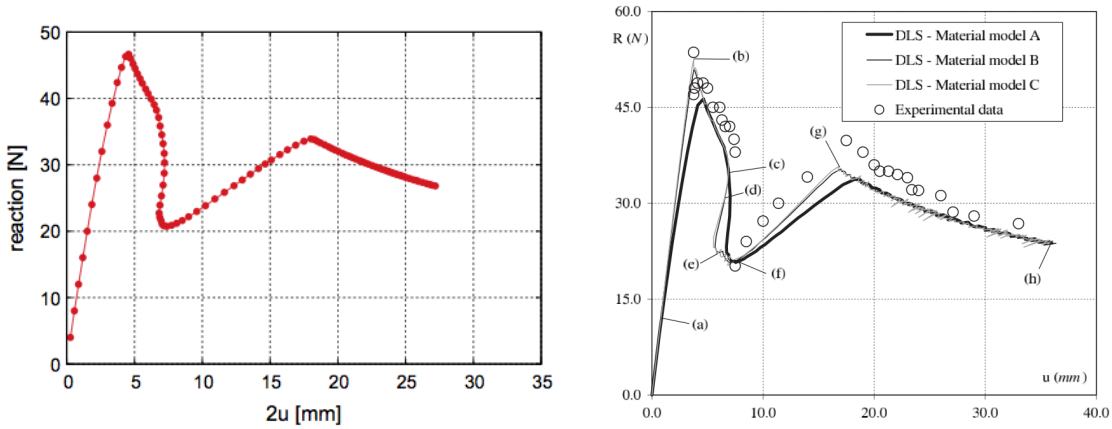


Figure 13: Multiple delamination problem: reaction-prescribed displacement curve (left) and reference solution [70] (right).

The deformed configuration obtained is presented in Fig. 14. The response of the specimen in terms of the reaction and two times the displacement at the top left corner is plotted in Fig. 13. A good agreement with the solution reported in [70] was obtained.

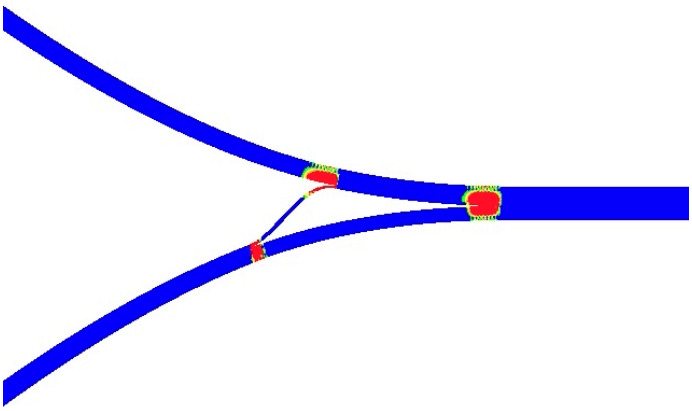


Figure 14: Multiple delamination analysis: deformed mesh.

5.3. Intergranular fracture of polycrystal

In this section the intergranular fracture of a specimen made of polycrystalline material is presented. The geometry of the specimen together with the FE mesh is shown in figure 15. The bulk material is simply an isotropic elastic one with $E = 72000$ N/mm^2 and $\nu = 0.33$. The interfaces are modelled using the Xu-Needleman exponential TSL [29] with the following parameters $\beta = 1$, r and q are taken of 0 and 1 respectively, $\sigma_{\max} = 500$ N/mm^2 and $\delta_n = \delta_t = 0.001$. The left edge is fixed in both directions and a horizontal force is applied on the right edge. The deformed configuration is given in Fig. 16. For a throughout study on inter-granular fracture of polycrystals, we refer to the work given in [71] which employs XFEM/GFEM.

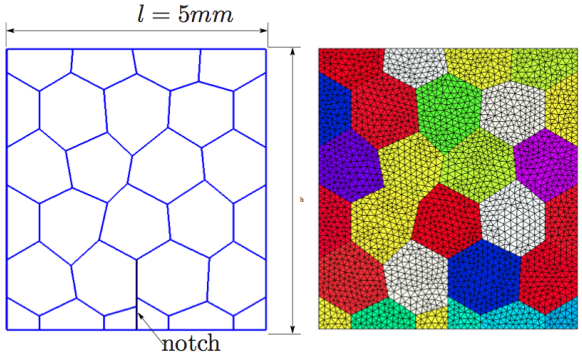


Figure 15: A pre-notched polycrystalline sample: geometry and FE discretization. Note that all grains are of the same material but assigned to different groups while being meshed in Gmsh so that the proposed algorithm can be applied.

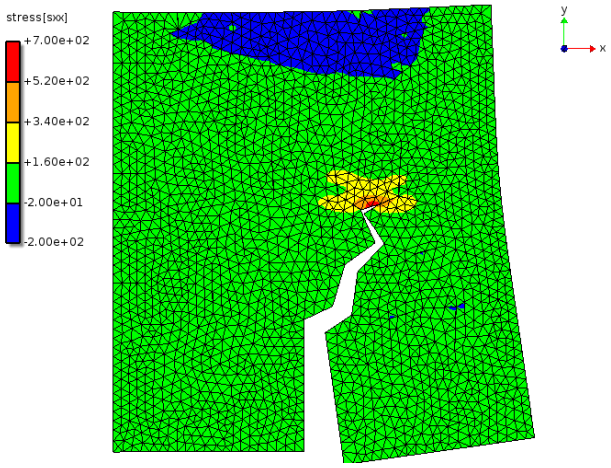


Figure 16: A pre-notched polycrystalline sample: deformed configuration.

5.4. Some 3D problems

Finally we present some examples on 3D fracture problems using the 2D interface elements as shown in Fig. 17. In the top-left figure is shown the deformation of a sample made of a composite material with four long fibers. The

sample is subjected to a uniaxial displacement in the horizontal direction. In the top-right figure, a thin slice of a polycrystalline solid was studied. Again the sample was under uniaxial tensile loading. The bottom figure is the familiar 3D cantilever beam with an initial notch at the middle plane of the beam starting from the left. It should be emphasized that the examples were provided to demonstrate the capability of the presented program to generate 2D cohesive elements for 3D geometries..

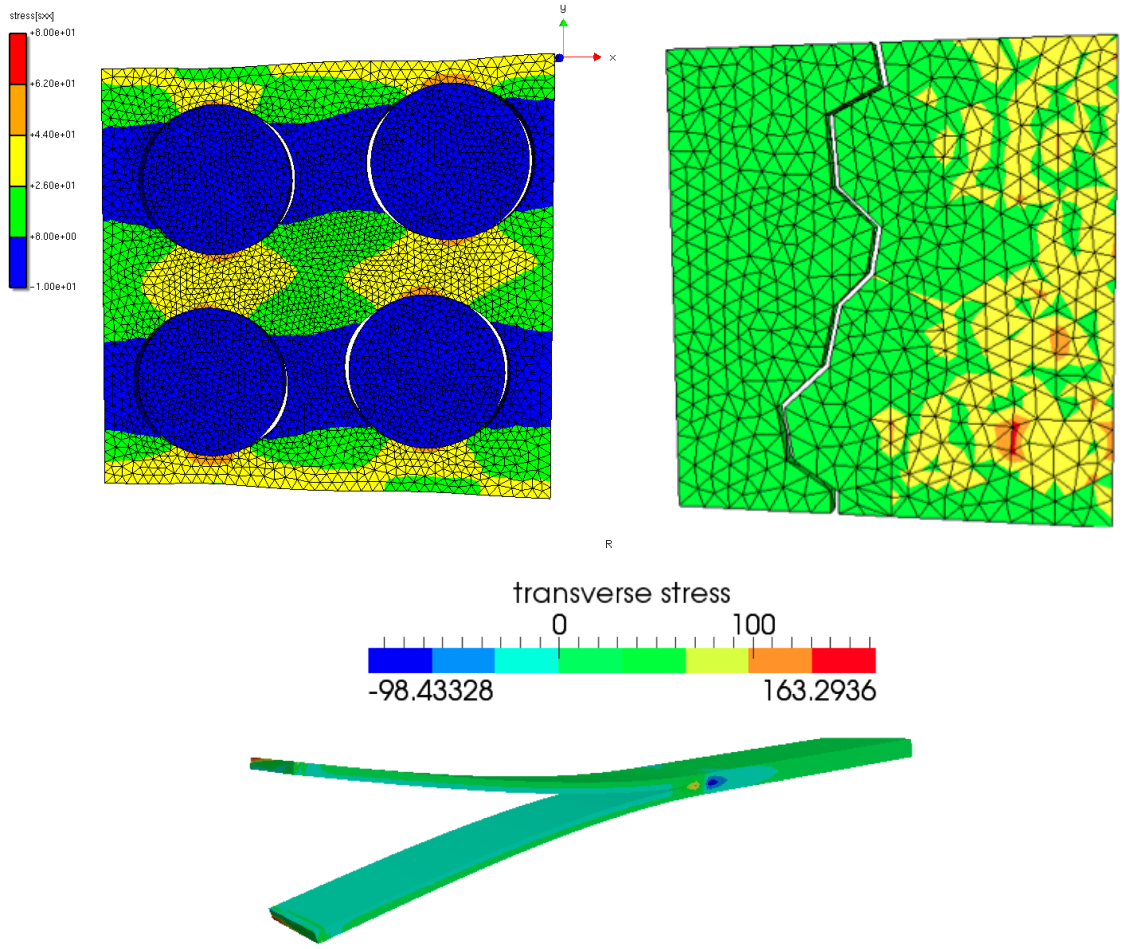


Figure 17: Three dimensional fracture analyses with 2D cohesive interface elements: material interface debonding, inter-granular fracture and delamination of composites.

5.5. Dynamic fracture simulation with dG method

A doubly notched specimen under an impact load is investigated. The geometry of the specimen is shown in Fig. 18, and the impact loading is applied by a projectile. In the experiment [72], two different failure modes were observed by modifying the projectile speed, v_0; at high impact velocities, a shear band is observed to emanate from the notch at an angle of 10^o with respect to the initial notch and at lower strain rates, brittle failure with a crack propagation angle

of about 70^o is observed. We are interested only in the velocity range that resulted in a brittle failure mode. Note that the failure model also depends on the notch tip radius, see [73] for a discussion on this. The material parameters of Maraging steel 18Ni(300), which are taken from [74], are as follows: Young's modulus $E = 190$ GPa ($\times 10^3$ MPa), a Poisson ratio of $\nu = 0.3$, and a density of $\rho = 8000$ kg/m^3. The material properties for the interface elements (the initially rigid damage-based bilinear cohesive law [68, 69] is used) are $\tau_1^0 = \tau_3^0 = 1733$ MPa (N/mm^2) (tensile and shear strength), $G_{Ic} = G_{IIc} = 22.2$ N/mm (fracture energies), $K = 10^6$ N/mm^3 and $\mu = 1.0$. The wave speed is $c = \sqrt{E/\rho} = 4873.4$ m/s. Due to symmetry only the upper half was modeled and the impact velocity v_0 of 16.54 m/s is applied along the edge of the sample hit by the projectile. In our implementation we imposed the displacement boundary conditions rather than the velocities. Therefore, on the impact edge, a prescribed displacement $u_x = v_0 \times t$ (t denotes the time) is imposed.

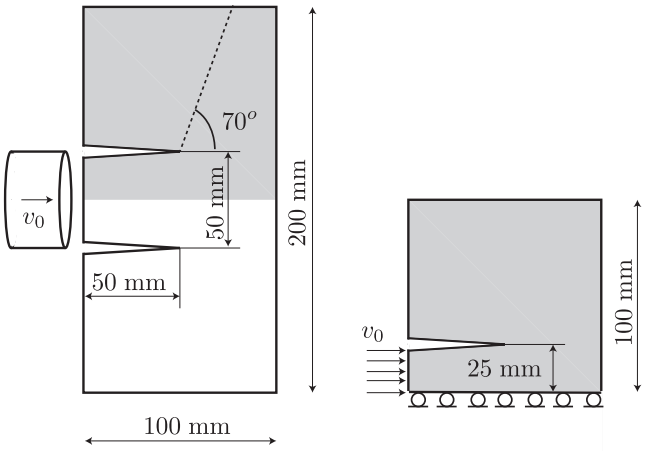


Figure 18: Experimental set up for edge-cracked plate under impulsive loading: shaded region denotes the numerically modeled region (left) and the plane strain FE model with boundary conditions (right). Also shown is the crack path (dashed line) experimentally observed.

The FE mesh is depicted in Fig. 19 with a refinement of the region where the crack is going to appear. The crack trajectory with respect to time is shown in Fig. 20. It can be seen that a good agreement of the crack path with the experiment was obtained. The propagation angle is estimated to be around 69^o, which agrees well with the experimental prediction (70^o). Although the overall crack path follows an inclined direction, from Fig. 20b, the initial crack propagation shows a short vertical segment. Other researchers [37, 73, 74] also reported similar results, although in [74] a very short vertical crack segment was present. In order to study mesh convergence of crack path, we performed another simulation with a slightly different mesh, called *mesh2* in the sequel (the other is called *mesh1*). The crack path obtained with *mesh2* is compared with the one obtained with *mesh1* and the result is depicted in Fig. 21. As can be seen, the crack paths are quite similar regardless of using different meshes.

6. Conclusions

In this paper, simple algorithms to generate cohesive surfaces into a finite element mesh was presented. The resulting program, which is available for download without any cost on our website, is believed to be useful for researches on fracture modelling of structure and material. The program is able to create both one and two dimensional cohesive

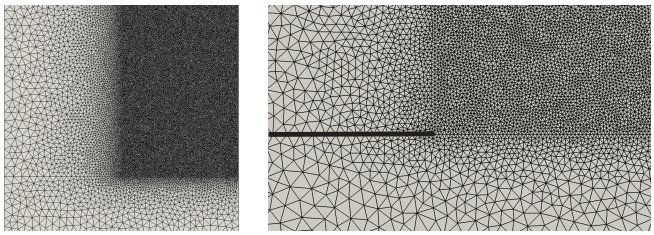


Figure 19: Edge-cracked plate under impulsive loading: unstructured mesh of three node triangle elements (46 832 bulk elements, 23 590 nodes) and zoom in of the region around the crack tip. The thick line denotes the initial crack. After inserting interface elements, the mesh consists of 135 118 nodes with 66 853 four node interface elements. Note that interface elements were not placed in the domain below the initial crack. The smallest element size (for unstructured meshes consisting of triangle/quadrilateral elements, h_e was computed as the ratio of the element area over the maximum edge length) is 0.27 mm which leads to the time step of 5×10^{-9} s ($0.1h_e/c$).

elements, both linear and quadratic. In the current implementation, only the mesh created by Gmsh is supported and the output format is tailored to our in-house FE code. Extension to other mesh formats, is, however, straightforward. Future work would be to incorporate this directly into a good free mesh generator like Gmsh.

Also given in the paper was the flowchart of a general interface element implementation that works for any interface element type in both 2D and 3D. This flowchart is interestingly very closed to the one of the phantom node method, a simple version of XFEM, that has simplified the implementation of XFEM into existing FE codes.

Concerning fracture modelling using interface elements, as mentioned in the introduction, one remedy to the reliance of the method on the use of a high dummy stiffness is to exactly constraint the interface elements and activate them when a certain failure criterion is fulfilled. This technique was adopted in [19] but without implementation details. It would be interesting to perform a comparative study between this technique and the discontinuous Galerkin based cohesive interface elements [45, 20] which approximately constraint the interface elements in the pre-failure stage.

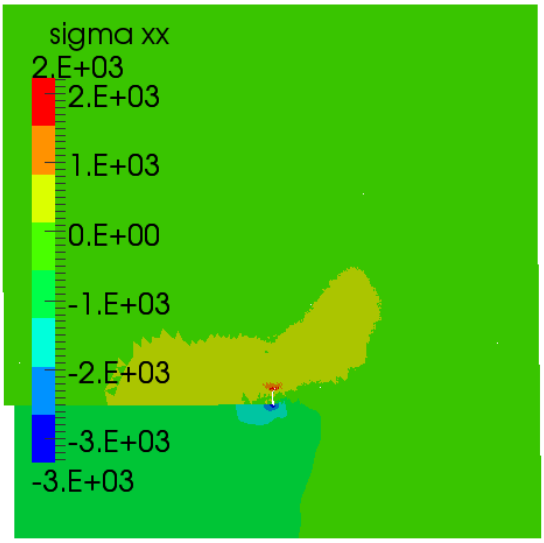
Acknowledgements

The author would like to thank Prof. Bert Sluys for the support during the author's PhD time in TU Delft and Dr. Clemens Verhoosel (Eindhoven University of Technology, The Netherlands) for the fruitful discussion on the preliminary ideas of the meshing algorithm. The input on grain topology from Dr. Zahid Sabir (former PhD student at Delft University of Technology, The Netherlands) is also gratefully acknowledged. The author would like to express the gratitude towards Drs. Erik Jan Lingen and Martijn Stroeven at the Dynaflow Research Group, Houtsingel 95, 2719 EB Zoetermeer, The Netherlands for providing us the numerical toolkit jem/jive. Financial support of the Framework Programme 7 Initial Training Network Funding under grant number 289361 "Integrating Numerical Simulation and Geometric Design Technology" is kindly acknowledged.

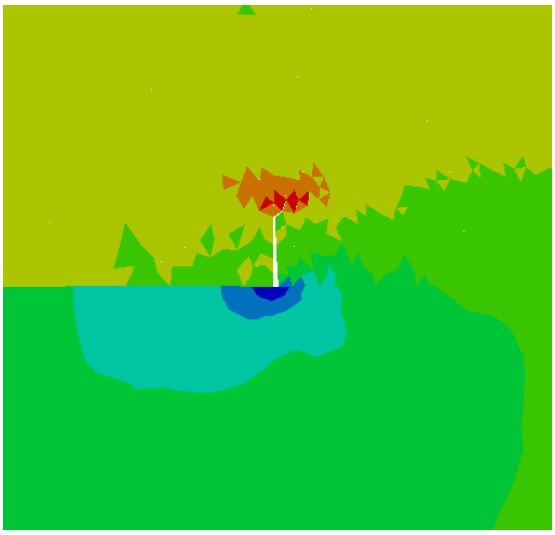
References

[1] D.S. Dugdale. Yielding of steel sheets containing slits. *Journal of the Mechanics and Physics of Solids*, 8(2):100–104, 1960.

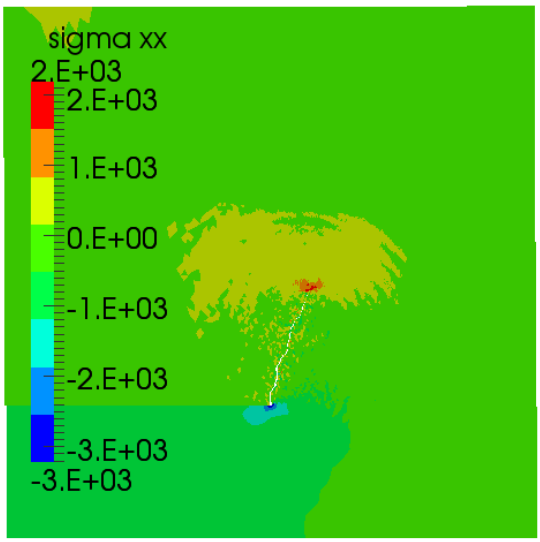
[2] G. Barenblatt. The mathematical theory of equilibrium cracks in brittle fracture. *Advanced Applied Mechanics*, 7:55–129, 1962.



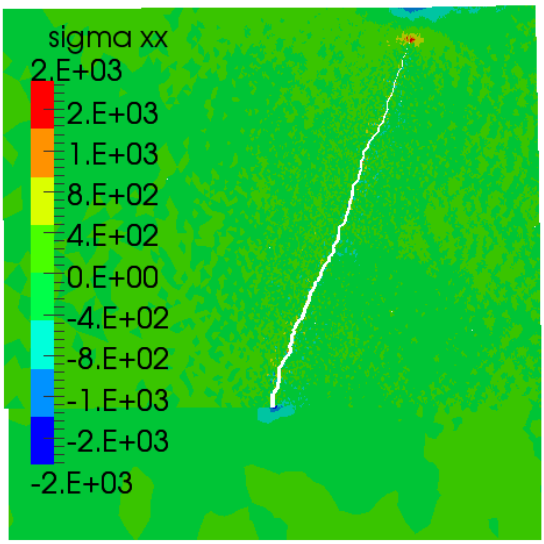
(a) $t = 26$ μs



(b) $t = 26$ μs, zoom

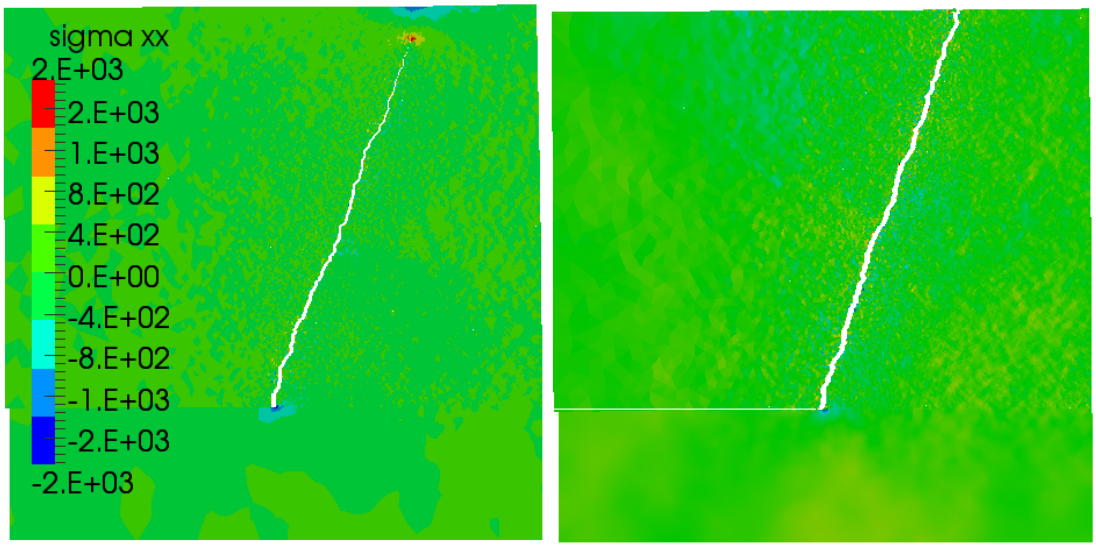


(c) $t = 40$ μs



(d) $t = 75$ μs

Figure 20: Edge-cracked plate under impulsive loading: crack path with respect to time. The unit of stresses is MPa.



(a) $t = 75$ μs, *mesh1* (b) $t = 100$ μs, *mesh2*

Figure 21: Edge-cracked plate under impulsive loading: crack paths obtained with two meshes.

[3] A. Hillerborg, M. Moder, 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*, 6(6):773–781, 1976.

[4] D. Dias-da Costa, J. Alfaiate, L.J. Sluys, and E. Júlio. A comparative study on the modelling of discontinuous fracture by means of enriched nodal and element techniques and interface elements. *International Journal of Fracture*, 161(1):97–119, 2010.

[5] T. Rabczuk. Computational methods for fracture in brittle and quasi-brittle solids: State-of-the-art review and future perspectives. *ISRN Applied Mathematics*, 2013.

[6] J. C. Simo, J. Oliver, and F. Armero. An analysis of strong discontinuities induced by strain-softening in rate-independent inelastic solids. *Computational Mechanics*, 12(5):277–296, 1993.

[7] F. Armero and C. Linder. Numerical simulation of dynamic fracture using finite elements with embedded discontinuities. *International Journal of Fracture*, 160(2):119–141, 2009.

[8] C. Linder and F. Armero. Finite elements with embedded branching. *Finite Elements in Analysis and Design*, 45(4):280 – 293, 2009.

[9] D. Dias da Costa, J. Alfaiate, L.J. Sluys, and E. Júlio. A discrete strong discontinuity approach. *Engineering Fracture Mechanics*, 76(9):1176 – 1201, 2009.

[10] J. M. Melenk and I. Babuška. The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139:289–314, 1996.

[11] N. Moës, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46(1):131–150, 1999.

[12] N. Sukumar, D.L. Chopp, N. Moës, and T. Belytschko. Modeling holes and inclusions by level sets in the extended finite element method. *International Journal for Numerical Methods in Engineering*, 190(47):6183—6200, 2001.

[13] N. Moës, M. Cloirec, P. Cartraud, and J.-F. Remacle. A computational approach to handle complex microstructure geometries. *Computer Methods in Applied Mechanics and Engineering*, 192(28-30):3163–3177, 2003.

[14] T. Hettich and E. Ramm. Interface material failure modeled by the extended finite-element method and level sets. *Computer Methods in Applied Mechanics and Engineering*, 195(37-40):4753–4767, July 2006.

[15] G. N. Wells and L. J. Sluys. A new method for modelling cohesive cracks using finite elements. *International Journal for Numerical Methods in Engineering*, 50:2667–2682, 2001.

[16] 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*, 84(3):253–304, 2010.

[17] P.M.A. Areias and T. Belytschko. Analysis of three-dimensional crack initiation and propagation using the extended finite element method. *International Journal for Numerical Methods in Engineering*, 63:760–788, 2005.

[18] A. Simone, C. A. Duarte, and E. Van der Giessen. A generalized finite element method for polycrystals with discontinuous grain boundaries. *International Journal for Numerical Methods in Engineering*, 67(8):1122–1145, 2006.

[19] C.V. Verhoosel and M.A. Gutiérrez. Modelling inter- and transgranular fracture in piezoelectric polycrystals. *Engineering Fracture Mechanics*, 76(6):742–760, 2009.

[20] L. Wu, D. Tjahjanto, G. Becker, A. Makradi, A. Jérusalem, and L. Noels. A micromeso-model of intra-laminar fracture in fiber-reinforced composites based on a discontinuous Galerkin/cohesive zone method. *Engineering Fracture Mechanics*, 104(0):162 – 183, 2013.

[21] 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*, 30(9):1239 – 1253, 1993.

[22] O. Allix, P. Ladevèze, and A. Corigliano. Damage analysis of interlaminar fracture specimens. *Composite Structures*, 31(1):61 – 74, 1995.

[23] 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*, 50(7):1701–1736, 2001.

[24] G.N.Wells. *Discontinuous modelling of strain localisation and failure*. PhD thesis, Delft University of Technology, 2001.

[25] J. J. C. Remmers, R. de Borst, and A. Needleman. A cohesive segments method for the simulation of crack growth. *Computational Mechanics*, 31(1):69–77, 2003.

[26] C.G. Dávila, P.P. Camanho, and A. Turon. Effective simulation of delamination in aeronautical structures using shells and cohesive elements. *Journal of Aircraft*, 45(2):663–672, 2008.

[27] A. Needleman. A continuum model for void nucleation by inclusion debonding. *Journal of Applied Mechanics*, 54(3):525–531, 1987.

[28] V. Tvergaard. Effect of fibre debonding in a whisker-reinforced metal. *Materials Science and Engineering: A*, 125(2):203 – 213, 1990.

[29] X.P. Xu and A. Needleman. Numerical simulations of fast crack growth in brittle solids. *Journal of the Mechanics and Physics of Solids*, 42(9), 1994.

[30] M. G. A. Tijssens, L. J. Sluys, and E. van der Giessen. Simulation of fracture of cementitious composites with explicit modeling of microstructural features. *Engineering Fracture Mechanics*, 68(11):1245–1263, 2001.

[31] I. Carol, C. M. López, and O. Roa. Micromechanical analysis of quasi-brittle materials using fracture-based interface elements. *International Journal for Numerical Methods in Engineering*, 52(1-2):193–215, 2001.

[32] A. Caballero, C.M. López, and I. Carol. 3D meso-structural analysis of concrete specimens under uniaxial tension. *Computer Methods in Applied Mechanics and Engineering*, 195(52):7182–7195, 2006.

[33] J. F. Molinari, G. Gazonas, R. Raghupathy, A. Rusinek, and F. Zhou. The cohesive element approach to dynamic fragmentation: the question of energy convergence. *International Journal for Numerical Methods in Engineering*, 69(3):484–503, 2007.

[34] R. Radovitzky, A. Seagraves, M. Tupek, and L. Noels. A scalable 3D fracture and fragmentation algorithm based on a hybrid, discontinuous Galerkin, cohesive element method. *Computer Methods in Applied Mechanics and Engineering*, 200(14):326 – 344, 2011.

[35] F. Cirak, M. Ortiz, and A. Pandolfi. A cohesive approach to thin-shell fracture and fragmentation. *Computer Methods in Applied Mechanics and Engineering*, 194(21–24):2604–2618, 2005.

[36] G. Becker, C. Geuzaine, and L. Noels. A one field full discontinuous Galerkin method for Kirchhoff–Love shells applied to fracture mechanics. *Computer Methods in Applied Mechanics and Engineering*, 200(4546):3223 – 3241, 2011.

[37] P.A. Klein, J.W. Foulk, E.P. Chen, S.A. Wimmer, and H.J. Gao. Physics-based modeling of brittle fracture: cohesive formulations and the application of meshfree methods. *Theoretical and Applied Fracture Mechanics*, 37(13):99 – 166, 2001.

[38] T. Rabczuk and T. Belytschko. Cracking particles: a simplified meshfree method for arbitrary evolving cracks. *International Journal for Numerical Methods in Engineering*, 61(13):2316–2343, 2004.

[39] T. Rabczuk and T. Belytschko. A three-dimensional large deformation meshfree method for arbitrary evolving cracks. *Computer Methods in Applied Mechanics and Engineering*, 196(2930):2777 – 2799, 2007.

[40] T. Rabczuk and G. Zi. A meshfree method based on the local partition of unity for cohesive cracks. *Computational Mechanics*, 39(6):743–760, 2007.

[41] S. Bordas, T. Rabczuk, and G. Zi. Three-dimensional crack initiation, propagation, branching and junction in non-linear materials by an extended meshfree method without asymptotic enrichment. *Engineering Fracture Mechanics*, 75(5):943 – 960, 2008.

[42] T. Rabczuk, P. M. A. Areias, and T. Belytschko. A simplified mesh-free method for shear bands with cohesive surfaces. *International Journal for Numerical Methods in Engineering*, 69(5):993–1021, 2007.

[43] Y. Doh Ha and F. Bobaru. Characteristics of dynamic brittle fracture captured with peridynamics. *Engineering Fracture Mechanics*, 78(6):1156 – 1168, 2011.

[44] A. Agwai, I. Guven, and E. Madenci. Crack propagation in multilayer thin-film structures of electronic packages using the peridynamic theory. *Microelectronics Reliability*, 51(12):2298 – 2305, 2011.

[45] J. Mergheim, E. Kuhl, and P. Steinmann. A hybrid discontinuous Galerkin/interface method for the computational modelling of failure. *Communications in Numerical Methods in Engineering*, 20(7):511–519, 2004.

[46] 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*, 74(10):1665–1682, July 2007.

[47] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal of Numerical Analysis*, pages 1749–1779, 2001.

[48] V.P. Nguyen, O. Lloberas-Valls, M. Stroeven, and L. J. Sluys. Computational homogenization for multiscale crack modelling. Implementational and computational aspects. *International Journal for Numerical Methods in Engineering*, 89(2):192–226, 2012.

[49] V. P. Nguyen and H. Nguyen-Xuan. High-order B-splines based finite elements for delamination analysis of laminated composites. *Composite Structures*, 102:261–275, 2013.

[50] F.P. van der Meer and L.J. Sluys. A numerical investigation into the size effect in the transverse crack tension test for mode II delamination. *Composites Part A: Applied Science and Manufacturing*, 54(0):145 – 152, 2013.

[51] T. Belytschko, W. K. Liu, and B. Moran. *Nonlinear Finite Elements for Continua and Structures*. John Wiley & Sons, LTD - ISBN:0-471-98774-3, 2003.

[52] M. Ortiz and A. Pandolfi. Finite-deformation irreversible cohesive elements for three-dimensional crack-propagation analysis. *International Journal for Numerical Methods in Engineering*, 44(9):1267–1282, 1999.

[53] J.H. Song, P. M. A. Areias, and T. Belytschko. A method for dynamic crack and shear band propagation with phantom nodes. *International Journal for Numerical Methods in Engineering*, 67(6):868–893, 2006.

[54] T. Rabczuk, G. Zi, A. Gerstenberger, and W. A. Wall. A new crack tip element for the phantom-node method with arbitrary cohesive cracks. *International Journal for Numerical Methods in Engineering*, 75(5):577–599, 2008.

[55] J. Mergheim, E. Kuhl, and P. Steinmann. A finite element method for the computational modelling of cohesive cracks. *International Journal for Numerical Methods in Engineering*, 63:276–289, 2005.

[56] J. C. J. Schellekens and R. de Borst. On the numerical integration of interface elements. *International Journal for Numerical Methods in Engineering*, 36(1):43–66, 1993.

[57] V. P. Nguyen, P. Kerfriden, and S. Bordas. Isogeometric cohesive elements for two and three dimensional composite delamination analysis. *Composites Part B: Engineering*, 2013. http://arxiv.org/abs/1305.2738.

[58] M. Anvari, I. Scheider, and C. Thaulow. Simulation of dynamic ductile crack growth using strain-rate and triaxiality-dependent cohesive elements. *Engineering Fracture Mechanics*, 73(15):2210 – 2228, 2006.

[59] C. Geuzaine and J. F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

[60] E. J. Lingen and M. Stroeven. Jem/Jive-a C++ numerical toolkit for solving partial differential equations. `http://www.habanera.nl/`.

[61] X. Liu, R. Duddu, and H. Waisman. Discrete damage zone model for fracture initiation and propagation. *Engineering Fracture Mechanics*, 92(0):1 – 18, 2012.

[62] M. A. Gutiérrez. Energy release control for numerical simulations of failure in quasi-brittle solids. *Communications in Numerical Methods in Engineering*, 20:19–29, 2004.

[63] 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*, 77(9):1290–1321, 2009.

[64] F. van der Meer and L. Sluys. A phantom node formulation with mixed mode cohesive law for splitting in laminates. *International Journal of Fracture*, 158(2):107–124, 2009.

[65] J.J.C. Remmers. *Discontinuties in materials and structures- a unifying computational approach*. PhD thesis, Delft University of Technology, 2006.

[66] G.T. Camacho and M. Ortiz. Computational modelling of impact damage in brittle materials. *International Journal of Solids and Structures*, 33(2022):2899 – 2938, 1996.

[67] G. Alfano and M. A. Crisfield. Solution strategies for the delamination analysis based on a combination of local-control arc-length and line searches. *International Journal for Numerical Methods in Engineering*, 58(7):999–1048, 2003.

[68] A. Turon, P.P. Camanho, J. Costa, and C.G. Dávila. A damage model for the simulation of delamination in advanced composites under variable-mode loading. *Mechanics of Materials*, 38(11):1072 – 1089, 2006.

[69] P. P. Camanho, C. G. Dávila, and M. F. de Moura. Numerical simulation of mixed-mode progressive delamination in composite materials. *Journal of Composite Materials*, 37(16):1415–1438, 2003.

[70] G. Alfano and M. A. Crisfield. Solution strategies for the delamination analysis based on a combination of local-control arc-length and line searches. *International Journal for Numerical Methods in Engineering*, 58:999–1048, 2003.

[71] Z. Shabir, E. van der Giessen, C. A. Duarte, and A. Simone. The role of cohesive properties on intergranular crack propagation in brittle polycrystals. *Modelling and Simulations in Materials Science and Engineering*, 19(3):035006, 2011.

[72] J. F. Kalthoff and S. Winkler. Failure mode transition at high rates of shear loading. *International Conference on Impact Loading and Dynamic Behavior of Materials*, 1:185–195, 1987.

[73] Z. Zhang and G. H. Paulino. Cohesive zone modeling of dynamic failure in homogeneous and functionally graded materials. *International Journal of Plasticity*, 21(6):1195 – 1254, 2005.

[74] 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*, 92(1):1–35, 2012.