3D Multiscale Fracture in Fiber-Reinforced Composites

Daniel Duarte Costa Farinha

TU Delft

3D Multiscale Fracture in Fiber-Reinforced Composites

by

Daniel Duarte Costa Farinha

to obtain the degree of Master of Science
in Aerospace Engineering
at the Delft University of Technology,
to be defended publicly on Tuesday March 14, 2023 at 14:00.

Student number:	5359325	
Submitted on:	February 15, 2023	
Thesis committee:	Dr. S.R. Turteltaub,	TU Delft, supervisor
	Dr. B. Giovanardi,	TU Delft
	Dr.ir. O. K. Bergsma,	TU Delft

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Abstract

Over the last years, there has been a growing presence of composite materials in aerospace applications. In many of these, having a good understanding of how damage appears and evolves in the materials used is of paramount importance. While this has been achieved quite successfully for metals, the same cannot be said for composite materials. This is a direct consequence of their highly complex fracture behaviour, which is heavily dependent on their microstructure. Multiscale methods may play a key role in solving this issue, since they are able to combine information from multiple scales, something that more traditional approaches can't.

A new multiscale framework for fracture is being developed at TU Delft for composite materials. This methodology condenses the fracture behaviour observed on a microscopic volume element into one or more effective macroscopic cracks, where such information is encoded in their properties, namely area, orientation and traction-separation response. This methodology has also been implemented in a Python tool. It performs the pre-processing and post-processing of the finite element analyses where the microscopic volume element is loaded until it fully cracks. Such finite element analyses are carried out using Abaqus.

In this thesis, this methodology and its implementation are extended to 3D in order to increase its versatility. This extension is both performed in terms of its analytical formulation and the numerical implementation of the pre-processing segment. In its current form, the tool is capable of generating FEM models for three-dimensional microscopic volume elements made up of multiple-ply layups with random spacial distributions of straight fibers within each ply. Due to limitations imposed by the meshing software used (Gmsh), microscopic volume elements with fiber orientations other than 0° and 90° could not be meshed. Furthermore, to properly implement Periodic Boundary Conditions (PBCs), a new step which was not present in 2D had to be added, where the mesh is corrected to meet the requirements for a proper implementation of PCBs.

During this thesis, it was not possible to implement and test the 3D extension of the post-processing segment. To deal with this, a simpler post-processing methodology was used, where only the bulk domain's response was taken into account. At the same time, in the scripts used for conventional post-processing where crack detection takes place, some proposed algorithms were implemented. These were not tested due to a lack of good data. Besides that, for those scripts where no changes were made but are needed in order to properly extend the conventional post-processing segment to 3D, a general description of possible algorithms is also presented.

A relatively low number of analyses were performed, due to the prohibitively high computational cost. Those performed focused on uniaxial and biaxial stretch load cases for both an unidirectional ply and for a [0,90,0] laminate. As for the results obtained, contrary to what would be expected, a strange behaviour was observed near the boundaries of the volume element where the load was applied for all analyses except one. Some possible causes for this problem were laid out. A thorough evaluation of all of them was not possible as a result of the high number of analyses needed and the associated computational cost. As such, this currently remains an open problem.

Acknowledgments

This thesis represents the end of 2 and a half years as a master's student at TU Delft, which has been an incredible experience that made me learn and grow a lot.

I would like to express my gratitude towards my supervisor Dr. Turteltaub for his help during this thesis. Over the many meetings we held during this period, he was always open to answer any questions and provide the required guidance through every hardship I faced.

Lastly, I would also like to thank all my family and friends for their support over these last few months.

Daniel Duarte Costa Farinha

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

AMR	Adaptive Mesh Refinement
CDM	Continuum Damage Mechanics
CZM	Cohesive Zone Model
ETSL	Effective Traction-Separation Law
FEM	Finite Element Method
FRP	Fiber Reinforced Plastic
GCG	Global Crack Group
HCM	Hard-Core Model
HCSM	Hard-Core Shaking Model
ICG	Internal Crack Group
IPSM	Initially Periodic Shaking Model
LEFM	Linear Elastic Fracture Mechanics
MPC	Multi-Point Constraint
NNA	Nearest Neighbour Algorithm
PBC	Periodic Boundary Condition
PUM	Partition of Unity Method
RSE	Random Sequential Expansion
RVE	Representative Volume Element
TPE	Total Potential Energy
TSL	Traction Separation Law
UD	Unidirectional
XFEM	eXtended Finite Element Method

Symbols

$()^{\mathrm{f}}$	Effective quantity
$\boldsymbol{\sigma}^{\mathrm{M}}$	Macroscopic stress tensor
$\boldsymbol{\varepsilon}$	Microscopic strain tensor
$\boldsymbol{\varepsilon}^{\mathrm{b}}$	Microscopic bulk strain tensor
$\boldsymbol{\varepsilon}^{\mathrm{f}}$	Microscopic fracture strain tensor
$\boldsymbol{m}$	Crack orientation

$\boldsymbol{n}$	Normal vector
δ	Opening displacement
δ^f	Critical opening displacement
$\delta^{\max}$	Maximum achieved opening displacement
δ_m^o	Opening displacement at damage initiation
δ_m	Total effective mixed-mode opening displacement
Γ	Fracture surface
Γ_J	J integral path
γ_s	Surface energy
$[\![\dot{\mathbf{u}}]\!]$	Crack opening rate
$[\![\mathbf{u}]\!]$	Crack opening
$\mathbb{C}$	Stiffness tensor
$\mathbf{F}$	Deformation gradient tensor
$\mathbf{f}_{\text{coh}}$	Cohesive traction-separation response function
$\mathbf{P}$	First Piola-Kirchhoff stress tensor
$\mathbf{t}$	Traction vector
$\mathbf{u}$	Displacement vector
$\mathbf{w}_m$	Microfluctuation field
$\mathbf{x}$	Position vector
μ	Viscous regularization parameter
ν	Poisson's ratio
Ω	Microsctructural domain
$\overline{\boldsymbol{\varepsilon}}$	Macroscopic strain tensor
$\partial\Omega$	Microsctructural domain's boundary
Π	Total potential energy
ρ	Radius of curvature at the crack's root
ρ_f	Fiber density
ρ_m	Matrix density
$\sigma, \boldsymbol{\sigma}$	Stress
σ_f	Critical applied stress
σ_Y	Yield stress
φ	PUM functions
a, b	Crack semi-axes
A	Area

a_{eff}	Irwin's effective crack length
$D, \boldsymbol{\kappa}$	Damage variable
E	Young's modulus
G	Energy release rate
G^f	Fracture toughness
J	J integral
K	Stress intensity factor
K_c	Fracture toughness
K_t	Stress concentration factor
l	RVE length
N	Shape functions
P^{b}	Bulk power
P^{ext}	Externally applied power
P^{f}	Fracture power
P^{M}	Macroscopic power
r_y, r_p	Plastic zone radius
t	RVE thickness
t^o	Ultimate traction
T^o_{eff}	Effective traction at damage initiation
v, V	Volume
V_f	Fiber volume fraction
W	Work
w	RVE width
w_e	Strain energy density
W_s	Work required to create new crack surfaces

1

Introduction

Despite the impacts of the pandemic, the aerospace industry is expected to keep growing in the next years [43]. As the industry grows, new planes will be acquired, both to accommodate this growth and also to replace older aircraft that are gradually retired. Over the last years, aircraft parts, which were traditionally made out of metal, have been increasingly replaced by composite parts, as these are more weight efficient and thus reduce the amount of fuel required for a given flight [33].

However, the use of composite parts has been hindered due to the still limited knowledge of their fracture behaviour. While failure in metals is mostly well understood, with well-established failure criteria, the same is not true for composite parts. Many failure criteria for composites have been proposed, but none have been able to fully model the fracture behaviour, as observed in the World-Wide Failure Exercise [41]. In turn, to ensure that safety is not compromised, large knockdown factors have to be used, limiting the weight savings that can be achieved by using composite materials instead of metals. The difficulty in establishing appropriate failure criteria for composites stems from the complicated failure mechanisms involved in their fracture process, as well as the way these interact with each other. As a result, when deriving these failure criteria analytically, one must make a trade-off between having criteria that take into account all failure mechanisms or criteria that don't use many complex parameters requiring lengthy and complex test campaigns.

Traditionally, multiscale methods have been used to study materials that have complex microstructures. The damage growth behaviour can also be studied using these methods. A novel multiscale methodology is being developed at TU Delft, which studies the onset and growth of damage in composite materials. This information is then condensed into effective traction-separation laws (ETSLs) which implicitly model the influence on the material's response of all phases present in its microstructure. Furthermore, this methodology can also be used to draw conclusions about how certain factors, such as manufacturing defects or extreme temperature conditions influence the material's fracture response.

This thesis project aims to build upon previous work on this methodology and extend it to 3D problems. If successful, not only would it remove the plane strain assumption made for 2D problems, but it would also widen the range of problems where this can be employed, such as those involving multiple fiber directions. From a long-term perspective, this project is a relevant step towards the overall research's objective, which is to create a more versatile way of studying the impact of small changes to a material's microstructure in the overall fracture behaviour.

This thesis report is structured as follows: in chapter §2, a literature review of the relevant parts of fracture mechanics and multiscale methods is done. In chapter §3, the analytical formulation of this new methodology, with the adaptions needed to be employed in 3D problems, is presented. After that, its numerical implementation is covered in chapters §4 and §5. The first of these two chapters focuses on the pre-processing segment, indicating the relevant details of the finite element models created and of the software used to create them. On the other hand, the second part is focused on the post-processing of the results, describing the alternative scheme that was used for this thesis due to reasons that will be later explained in more detail, but also laying out suggestions for possible ways to implement the conventional scheme in 3D. The results obtained from the simulations performed are presented and discussed in chapter §6. Lastly, in chapter §7, the conclusions and recommendations are presented.

2

Literature Review

2.1. Fracture Mechanics

Fracture mechanics is the field concerned with predicting how and when do structures fail, by analyzing the mechanisms behind damage propagation in them.

2.1.1. Linear Elastic Fracture Mechanics (LEFM)

Linear Elastic Fracture Mechanics is the most simple and oldest branch of fracture mechanics. It assumes that the material obeys Hooke's law until it fails.

Early Developments

For a material to break, the chemical bonds that hold its atoms together need to be broken. Determining how much force would be required to move an atom in a bond from its equilibrium position to an infinitely long distance away provided a first insight into how much force was required for a material to break. However, such estimate was found to be 3 to 4 times higher than what was observed through experiments. At the time, Inglis [44] had already derived an expression for the stress concentration factor, K_t, at the tip of an elliptical crack with semi-axes a and b, which was:

$$K_t = 1 + 2\frac{a}{b} = 1 + 2\sqrt{\frac{a}{\rho}} \tag{2.1}$$

where ρ is the radius of curvature at the root [85]. Based on this expression, the discrepancy was attributed to imperfections within the material's structure that create localized stress concentrations. However, this expression also posed a problem: for a sharp crack ($b = 0$), the stress at the crack tip would be infinite, which meant that failure would occur as soon as an infinitesimal stress was applied. Such problem served as motivation for Griffith's work, which opted to establish energy-based requirements for crack growth instead of stress-based ones [1].

Griffith criteria

Griffith [35] derived its criteria through an energy balance. When a crack grows, new fracture surfaces are created. Therefore, in order for the crack to grow, the total energy must either remain constant or decrease. For an incremental increase in crack area $\mathrm{d}A$, the critical energy point for crack growth can be expressed as:

$$\frac{\mathrm{d}\Pi}{\mathrm{d}A} + \frac{\mathrm{d}W_s}{\mathrm{d}A} = 0 \tag{2.2}$$

where W_s is the work required to create the new crack surfaces and Π is the total potential energy (TPE) which includes the internal strain energy and work of external forces [1].

For an infinitely wide plate with a through thickness sharp crack, the critical (applied) stress value, σ_f, for which the crack grows, under plane stress conditions, is given by:

$$\sigma_f = \sqrt{\frac{2E\gamma_s}{\pi a}} \tag{2.3}$$

and under plane strain conditions by:

$$\sigma_f = \sqrt{\frac{\pi E \gamma_s}{2(1-\nu^2)a}} \tag{2.4}$$

where E is the Young's modulus of the material, γ_s the surface energy of the material, ν the Poisson's ratio of the material and a the crack's half-length.

Energy Release Rate

Irwin [45] further expanded on this by introducing the energy release rate G, defined as

$$G = -\frac{\mathrm{d}\Pi}{\mathrm{d}A} \tag{2.5}$$

which is more convenient to work with. Likewise, the crack only grows if the energy release rate is bigger than a critical value G^f, the fracture toughness. Although easier to deal with, this relation is however only valid in very specific cases. Besides the formation of the new crack surfaces, other underlying mechanisms might be present that absorb strain energy as well, such as the formation of a plastic region ahead of the crack tip in metals. Therefore, for it to be a valid approach, such non-linear effects cannot be present and plasticity can only take place on a very small region near the crack tip, i.e. only small scale yielding can occur [1].

Stress Intensity Factor

Another approach to deal with the singularity at the crack tip consists in introducing the concept of a stress intensity factor, K_{mode} (for each crack opening mode shown in Figure 2.1). It is extracted from the analytical solution of the stress field around the previously mentioned sharp crack and is given by:

$$\lim_{r\to 0} \sigma_{ij}^{(I)} = \frac{K_I}{\sqrt{2\pi r}} f_{ij}^{(I)}(\theta) \tag{2.6a}$$

$$\lim_{r\to 0} \sigma_{ij}^{(II)} = \frac{K_{II}}{\sqrt{2\pi r}} f_{ij}^{(II)}(\theta) \tag{2.6b}$$

$$\lim_{r\to 0} \sigma_{ij}^{(III)} = \frac{K_{III}}{\sqrt{2\pi r}} f_{ij}^{(III)}(\theta) \tag{2.6c}$$

where r and θ are the polar coordinates of a given point with respect to the crack tip and $f_{ij}^{(III)}(\theta)$ is the solution part that only depends on θ.

The stress intensity factor approach can also be used as a failure criterion. For a given stress applied, the stress field in the surrounding area of the crack is fully characterized by this parameter. Hence, if the stress intensity factor exceeds the critical value for a given mode (K_{Ic}, K_{IIc} and K_{IIIc}), the crack will grow until ultimate failure is observed. These parameters are also referred to as fracture toughness, although the units here are of stress$\cdot\sqrt{\text{length}}$ while G^f has units of energy/length [1].

The stress intensity factor and the energy release rate are analogous concepts, since they both describe the available energy for crack growth [78]. As a result, they can also be described in terms of one another. For

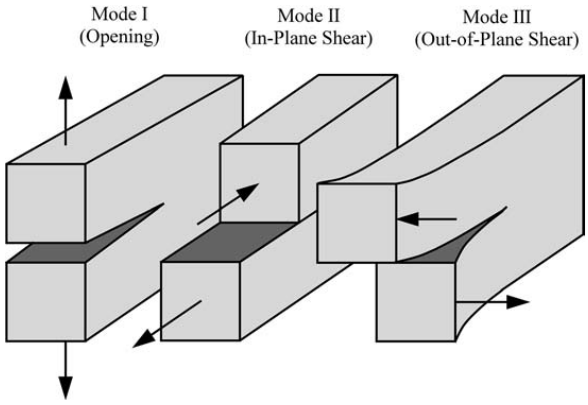


Figure 2.1: The three opening modes of a crack.[1]

a pure mode-I loading, they are related by the following expression:

$$G = \frac{K_I^2}{E'} \tag{2.7}$$

and for a mixed mode loading by:

$$G = \frac{K_I^2}{E'} + \frac{K_{II}^2}{E'} + \frac{K_{III}^2}{\mu} \tag{2.8}$$

where μ is the shear modulus, and $E' = E$ for plane stress or$E' = E/(1-\nu^2)$ for plane strain [1]. This relation is particularly useful, since G is easier to measure through experiments, while K is more convenient to use in the design of structures.

J Integral

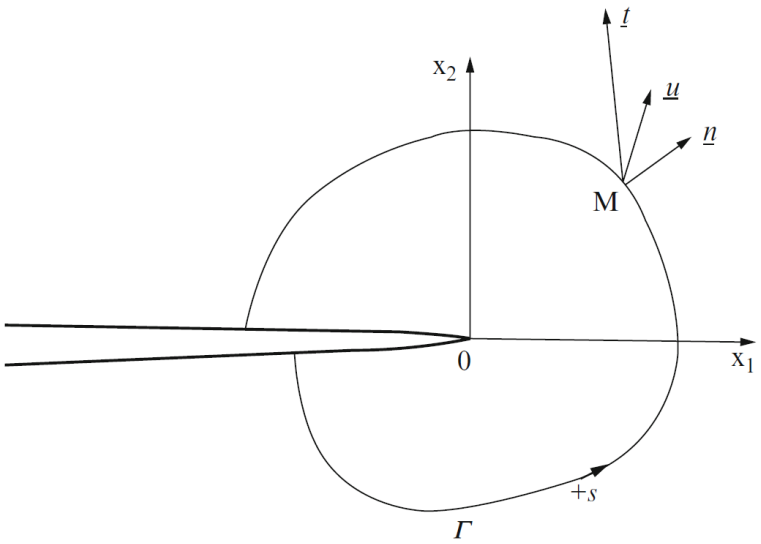


Figure 2.2: Schematic of a possible path for the J integral and the quantities associated to it.[27]

Introduced by Rice [80], the J integral extends the concept of G for a non-linear material. Assuming that the crack grows in the same direction and that the crack surfaces are traction-free, for a path Γ_J (Figure 2.2) it is defined as

$$J = \int_{\Gamma_J} \left(w_e \mathrm{d}x_2 - \mathbf{t}\frac{\partial \mathbf{u}}{\partial x_1}\mathrm{d}s \right) \tag{2.9}$$

where w_e is the strain energy density, $\mathbf{t}$ is the traction vector at a given point along the path for an outward pointing normal $\mathbf{n}$ and $\mathbf{u}$ is the displacement vector. It can be proved that for linear elastic problems $J = G$ and that the result is independent of the chosen path, as long as no unloading is performed if plasticity is present [1, 27].

Irwin plastic zone correction

As mentioned before, plasticity can lead to discrepancies between theory and the actual behaviour of materials, since not all the absorbed energy is spent on creating the new crack surfaces. To compensate for this without having to explicitly include models for plastic behaviour, Irwin proposed both a way to estimate the size of the region where it happens and a way of correcting the theory while taking this effect into account in an implicit manner [1].

Regarding its size, a first approximation consists of taking the solution for the stress field around the crack tip and defining such zone as the region where the stress is higher than the yield strength of the material (σ_Y). The size of this area is approximately:

$$r_y = \frac{1}{2\pi}\left(\frac{K_I}{\sigma_Y}\right)^2 \tag{2.10}$$

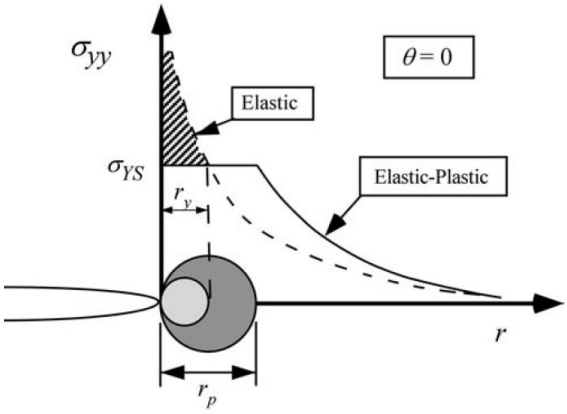


Figure 2.3: Comparison of the two methods to estimate the size of the plastic zone ahead of the crack tip and the force distribution assumed by each of them. [1]

This underestimates the size of the plastic zone, since there's a redistribution of the load when the material yields, which creates a bigger plastic area than what equation 2.10 predicts (Figure 2.3). If instead, a balance of forces is made along the boundary of the plastic zone, the following expression is obtained:

$$r_p = \frac{1}{\pi}\left(\frac{K_I}{\sigma_Y}\right)^2 \tag{2.11}$$

Lastly, Irwin introduced the effect of this region in the crack growth analysis procedures by defining an effective crack length, given by:

$$a_{eff} = a + r_y \tag{2.12}$$

2.1.2. Cohesive Zone Models

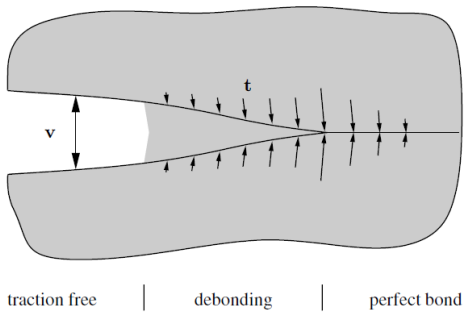


Figure 2.4: Schematic of the cohesive zone. Here, **v** represents the relative displacement δ. [17]

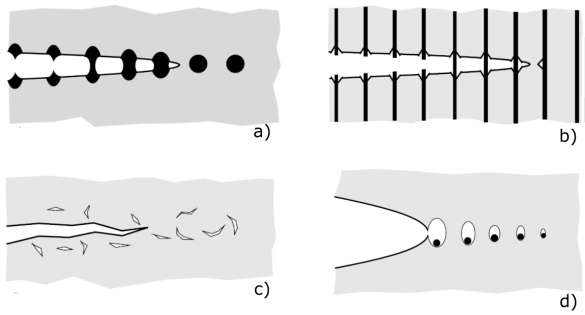


Figure 2.5: Some examples of degradation mechanisms: a) metal-particle reinforced ceramics b) fiber breaking in a FRP c) microcracks in a brittle material d) void nucleation and coalescence in a ductile metal. Adapted from [36]

Initially presented by Barenblatt [3] for brittle materials and by Dugdale [22] for ductile metals, a cohesive zone model (CZM) proposes the addition of a region in the wake of the crack tip (as shown in Figure 2.4), characterized by a relation between the traction $\mathbf{t}$ and the relative displacement $\boldsymbol{\delta}$ that models the degradation mechanism (Figure 2.5). Such relation is the traction separation law (TSL) and, by definition, all of them should verify the following relation:

$$G^f = \int_0^{\delta^f} t(\delta)\mathrm{d}\delta \tag{2.13}$$

where δ^f is the critical crack opening displacement, after which the cohesive stress becomes zero. Furthermore, TSLs usually have an internal damage variable that monitors and enforces the irreversibility of damage [17, 36].

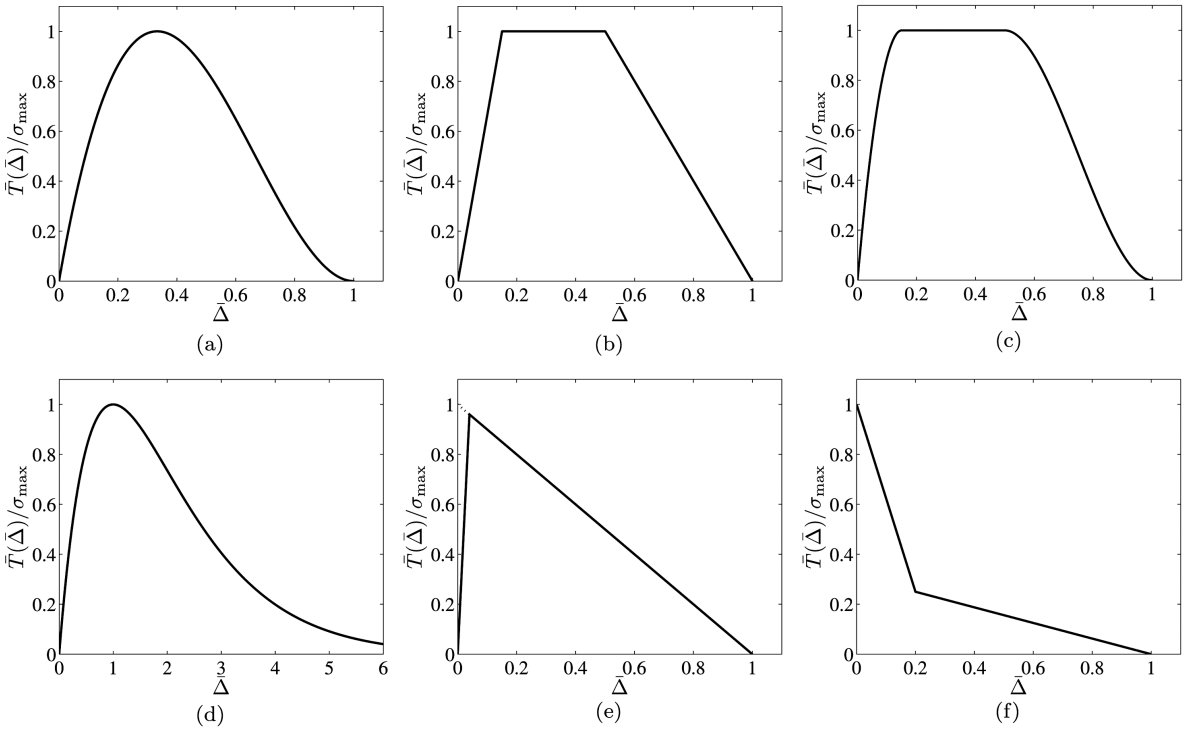


Figure 2.6: Some TSL examples. a)cubic polynomial, (b) trapezoidal, (c) smoothed trapezoidal, (d) exponential, (e) linear softening, and (f) bilinear softening. Adapted from [75]

One of the strengths of CZMs is that, besides modeling the growth of damage, they can also predict crack nucleation. Most of the theories mentioned so far fail to do this, since they already assume the presence of a crack in the material and only focus on evaluating if the necessary conditions for its growth are met [23].

The various TSLs found in the literature can be separated into nonpotential-based (or displacement-based) and potential-based TSLs.

Nonpotential-based TSLs define a TSL such as the ones presented in Figure 2.6 for each loading mode. For mixed-mode loadings, an effective TSL is constructed by combining the individual TSLs of each mode (Figure 2.7). To do so, it's necessary to specify, based on each individual contribution, criteria for the damage onset and evolution. For the damage onset, they can be based on one of the individual components reaching a threshold value or a combination of all of them reaching a given value (such as in a quadratic criterion). In the evolution, these can be divided into displacement-based and energy-based. In displacement based evolution, the increase in the damage variable is controlled by the total effective mixed-mode displacement, which is defined as:

$$\delta_m = \sqrt{\langle \delta_n \rangle^2 + \delta_t^2 + \delta_s^2} \tag{2.14}$$

with the subscripts s and t denoting the two tangential directions and the operator $\langle\cdot\rangle$ being the Macaulay's operator ($\langle\cdot\rangle$=max($\cdot$,0)). In energy-based evolution, the damage variable's increase is controlled by the ratio of energy released by each opening mode with respect to the total fracture toughness. Some methods for specifying the contribution of each mode in the total released energy include the power-laws and the Benzeggagh-Kenane criterion. Although very simple and easy to implement, nonpotential-based TSL have their main drawback in not guaranteeing consistency, since they aren't able to account for all possible opening paths [7, 12, 18, 75].

On the other hand, potential-based TSLs define a potential $\Psi(\delta_n, \delta_t)$ (for 2D but the same can be done for 3D) from which the traction components can be obtained through

$$t_n = \frac{\partial \Psi}{\partial \delta_n} \tag{2.15a}$$

$$t_t = \frac{\partial \Psi}{\partial \delta_t} \tag{2.15b}$$

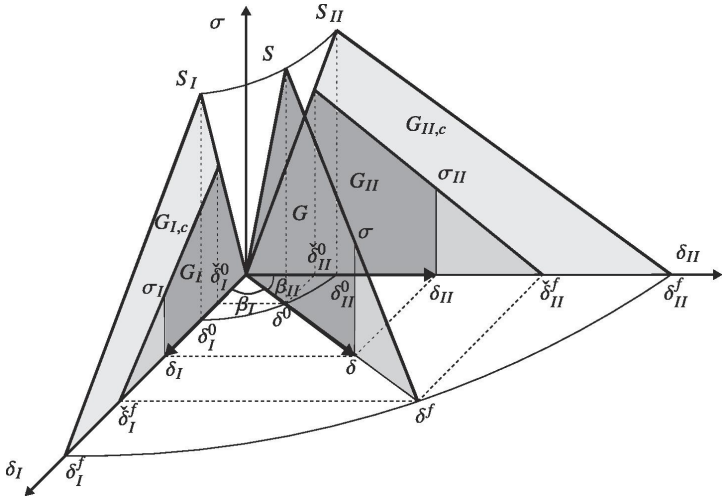


Figure 2.7: Example of nonpotential-based bi-linear TSL for mixed-mode I/II loading.[18]

Although they solve the consistency problem of nonpotencial-based TSLs, they also have some of their own. From the way they are defined, they are required to satisfy the symmetry condition of $\frac{\partial t_t}{\partial \delta_n} = \frac{\partial t_n}{\partial \delta_t}$, which may lead to overly complicated models. Moreover, some models might exhibit some inaccurate behaviour for specific loadings, as the one shown in Figure 2.8 does with the tangential traction for large shear loads. Furthermore, they also require more parameters to define, some of which might not be easy to experimentally measure.

In summary, both types of laws have shortcomings and the selection for a given problem must be made with great care.

Additionally, TSLs can be divided into extrinsic and intrinsic. In short, extrinsic TSLs have a non-zero traction value for $\delta = 0$ (such as the one shown in Figure 2.6 (f)), while intrinsic ones don't. This distinction has mostly implications regarding the numerical implementation, which will be discussed later. [75]

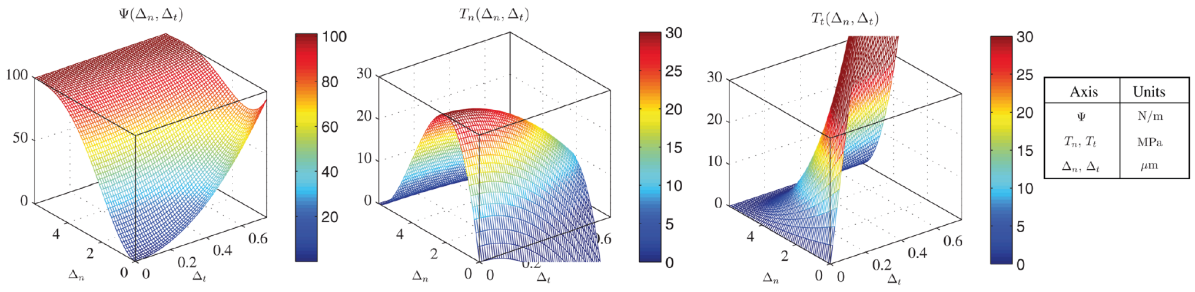


Figure 2.8: Example of a potential based TSL introduced by Needleman[70]. Adapted from [75]

2.1.3. Fracture of composite materials

Fracture in composite materials is a complex phenomenon, which is a direct result of their orthotropic nature and microscale heterogeneity.

While fracture in metals is remarkably well understood, the same cannot be said for composite materials. Metals are usually characterized by only having a single failure mechanism, requiring only the evaluation of the stress state to predict with acceptable confidence when and where a sample will fail. On the other hand, the multiple failure mechanisms that can occur in a composite material and these failure mechanisms are themselves coupled to the loading conditions as well. Furthermore, environmental conditions and the presence of small manufacturing defects can also have a great impact on their response. Such behaviour and its yet incomplete understanding by researchers makes it so that the application of composites in aerospace projects involves large knockdown factors, often undermining possible mass savings.

The way composites fail depends, not only on the stress state at a given point in the material, but also

on its orientation with respect to the constituents. For FRPs (Fiber-Reinforced Plastics) this is usually defined with respect to the fiber's directions. Considering the longitudinal direction to be the axial direction of the fibers, while transverse directions are those perpendicular to it, the failure mechanisms behind intra-ply failure of UD (unidirectional) composites for each combination of stress state and relative direction can be summarized as follows:

- **Longitudinal tension**: failure occurs through separated fiber breaks, which eventually link through a matrix crack [11].
- **Longitudinal compression**: failure usually occurs as a result of kink band formation, which develops from matrix shear instability caused by local fiber misalignments. It has also been observed, albeit on notched specimens, shear failure of the fibers instead of the formation of a kink band [38].
- **Transverse tension**: failure occurs through fiber-matrix debonding at the top and bottom regions of individual fibers, that coalesces and forms a matrix crack perpendicular to the loading direction[106]. Here, the top direction refers to the loading direction.
- **Transverse compression**: failure begins with fiber-matrix debonding at the lateral regions of individual fibers, which grow and link through matrix cracks until a peak stress is reached. Then, some of these cracks coalesce and form shear bands at a given angle determined by the material system [34, 106].
- **Longitudinal shear**: failure begins with fiber-matrix debonding, which then evolves into matrix failure. The formed fracture plane has an angle with respect to the longitudinal direction [108].
- **Transverse shear**: failure occurs in a similar mechanism as in transverse tension, but the final crack grows in the direction perpendicular to the direction of maximum principal stress [62, 101].

No literature was found on 3D failure mechanisms occurring within the individual plies of unidirectional composites. However, it is known that matrix cracks can be deflected by plies of different orientations and that different delamination cracks can join through matrix cracks [48, 49, 56].

2.2. Computational Fracture Mechanics

Classical FEM (Finite Element Method) interpolates nodal quantities, such as displacements, through shape functions, which are defined by continuous polynomials. However, fracture is a source of discontinuities, calling for the development of new techniques that incorporate the theoretical approaches discussed so far. In this section, the way these are numerically implemented will be quickly reviewed. Special emphasis is placed on cohesive elements, which is the technique that will be used in this thesis, but other ones will also be mentioned as possible alternatives.

2.2.1. Adaptive Mesh Refinement

Adaptive Mesh Refinement (AMR) approaches fracture by generating a new mesh such that no discontinuities are present in the newly meshed region. When such a discontinuity is predicted to appear, an algorithm repositions the nodes such that no interpolation is performed over the discontinuity. Once the new mesh has been defined, internal variables such as damage variables are projected into the new mesh by a mapping algorithm. After that, the load step is repeated using the new mesh and only then is the load incremented. The need to map quantities from one mesh to another can also be avoided if the analysis is repeated from the beginning [77].

This method was one of the first approaches to modeling fracture computationally and it's not used very often because it can be quite inefficient.

2.2.2. Cohesive Elements

Cohesive elements are one of the most widely used techniques. They implement CZMs through special interface elements that are embedded between bulk elements, as shown in Figure 2.9. Their behaviour is governed by a TSL and a detailed overview of the classic formulation of cohesive elements can be found in Nguyen [74]. Despite many shortcomings that stimulated the development of new techniques for fracture modelling, they are still widely used today and present in most FEM software packages.

Earlier, it was mentioned that TSLs can be classified as intrinsic or extrinsic, if they have a non-zero or zero traction value at $\delta = 0$, respectively. For intrinsic TSLs, the cohesive elements are added to a model prior

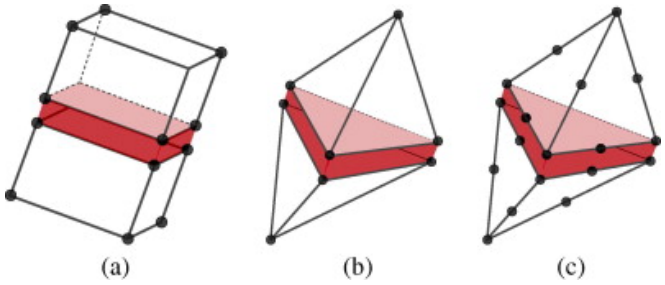


Figure 2.9: Some types of cohesive 3D elements: (a) linear brick, (b) linear tetrahedral and (c) quadratic tetrahedral elements. Cohesive elements, highlighted in red, are of initially zero thickness. [91]

to the analysis. The segment of the TSL up to the damage onset point introduces extra compliance into the model. Such an effect can be mitigated by using a very high penalty stiffness value in this segment, but this results in an ill-conditioned tangent stiffness matrix. In most applications, a compromise is made between both constraints by choosing a sufficiently high penalty stiffness value to limit the extra compliance added but still small enough to have limited numerical impact [73]. In applications such as dynamic crack growth, where this effect has implications in other parameters (e.g speed of sound), additional techniques such as rescaling the penalty stiffness during the simulation can further reduce the effect of the added compliance [24]. On the other hand, extrinsic TSLs are characterized by not having this segment. However, cohesive elements governed by them cannot be included in the same way in the model, thus requiring extra procedures that lead to an increase in the computational cost. In general, these consist in checking for a failure criterion and once it's met, adding cohesive elements in that interface [6]. This requires the use of complex data structures since the mesh has to be continuously changed to include the new elements, greatly increasing the complexity of the analysis. Another possibility consists in using intrinsic cohesive elements and having multi-point constraints (MPCs) restraining the opening of the element before the damage onset, which are selectively released as soon as damage appears [104]. The same can be accomplished by using a hybrid discontinuous Galerkin method [63, 73].

Furthermore, under changing mixed-mode conditions, restoration of the cohesive state can occur for some TSLs, whether they are nonpotential [83](Figure 2.10) or potential based [76]. This effect can be controlled by monitoring the energy released during the analysis, since this behaviour violates thermodynamical principles [83, 95]. In Abaqus, this is prevented by only allowing the damage variable to remain constant or increase.

Although very commonly used in problems where the crack path is known *a priori*, such as delamination in composites [8, 109], for problems where it isn't, the fact that they are only inserted between bulk elements represents a considerable drawback as it creates a severe mesh dependency. This can be mitigated through mesh refinement and the use of a random mesh, but as a result the computational cost can become significant for larger problems [73].

Another important aspect to consider when using cohesive elements is the integration scheme that will be used. Newton-Cotes and Gaussian integration are two classical integration schemes which are often employed. Newton-Cotes is more stable and provides more accurate nodal stress values. The presence of integration points at the vertices of the element also prevents interpenetration [84]. Gaussian integration, however, can induce spurious oscillations on the traction field, but outperforms Newton-Cotes for some

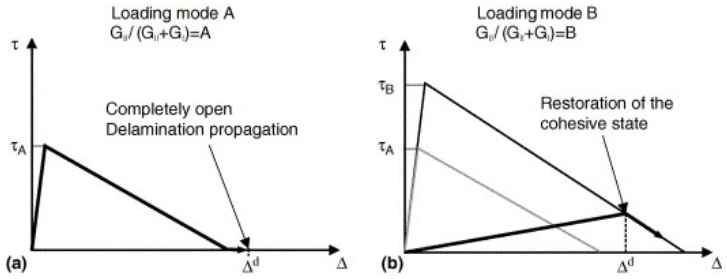


Figure 2.10: Example of restoration of the cohesive state in a nonpotential based TSL. [95]

applications [19] In terms of the number of integration points to be used, Do et al. [19] also observed that Newton-Cotes works best with 3 while for Gaussian integration 2 are enough. Some other integration methods with improved performance and stability that are also capable of circumventing the restriction that the cohesive element size should be smaller than 1/3-1/5 of the cohesive zone's length have also been developed [19]. For certain problems or more sophisticated element formulations, Gaussian integration can outperform Newton-Cotes, as seen by Russo and Chen [82].

Cohesive elements also display some unstable behaviour. As illustrated by Hamitouche et al.[39] for a 1D beam problem, fracture problems modeled with cohesive elements can, under specific circumstances, exhibit snap-back during damage nucleation, making convergence using the Newton-Rapson method difficult or even impossible. This behaviour originates from the sudden release of energy from the bulk elements into the cohesive elements. In order to correct this problem, many techniques were proposed, some of which are presented by Chaboche et al. [9]. The most widely used method is viscous regularization, which consists in adding a viscosity term to limit the energy release during unstable debonding, such that there is a stable equilibrium path connecting the stable states before and after damage nucleation [28]. Similarly to the problem posed by the penalty stiffness value, the amount of viscosity added must also be selected carefully. If the value is too small, the viscosity won't be enough to "transform" the unstable segment of the equilibrium path into a stable one. On the contrary, if the value is too big, the numerical accuracy will be affected by excessive dissipation. There are many viscous regularization schemes but, for reference, the one used in Abaqus is presented here [13]. With D_v being the degradation variable, μ the viscous regularization parameter and D the degradation variable evaluated in the inviscid backbone model, the scheme is implemented as:

$$\dot{D}_v = \frac{1}{\mu}(D - D_v) \tag{2.16}$$

Lastly, specialty elements have been developed to include more complex fracture behaviour in the formulation. Some examples include a 3D separable cohesive element to model the interaction between delamination cracks and matrix cracks developed by Lu et al. [56] or a finite thickness cohesive element for an accurate fracture energy dissipation modeling in thick adhesives, developed by Sarrado et al. [83].

2.2.3. Nodal Enrichment Methods: XFEM and its variations

Nodal Enrichment Methods deal with the aforementioned discontinuities by introducing in the nodal displacements, u, enrichments based on the partition of unity method (PUM)[2, 59, 60]. There are many different nodal enrichment methods, each differing in either what enrichment is used or how it is implemented, with XFEM (eXtended Finite Element Method) being the most notorious one. Initially introduced by Belytschko and Black [4], the basic XFEM method uses the following enrichment:

$$u(x) = \underbrace{\sum_{\forall I} N_I(x)\, u_I}_{\text{Classical FEM}} + \underbrace{\sum_{J \in S_\Gamma} N_J(x) \varphi_J(x) q_J}_{\text{Enrichment}} \tag{2.17}$$

where $N(x)$ denotes the shape functions, $J \subset I$ the enriched nodes $\varphi_J(x)$ the PUM functions and S_Γ the element domain crossed by the crack.

Within XFEM, there are two different forms of modelling the stress field around the crack tip: one is based on LEFM and the other on CZM. The approach based on LEFM makes a distinction between elements fully crossed by the crack and elements partially crossed by the crack, which contain the crack tip. For the nodes of the latter, the enrichment applied to nodes in elements fully crossed by the crack is replaced by a new one capable of representing the crack tip singularity. The approach based on CZM employs cohesive segments [79]. This method inserts a line segment when a failure criterion is met, that cuts through the element, passing over an integration point, and through the neighbouring elements. The opening behaviour is then governed by a cohesive law [4, 65].

One of the standout features of XFEM is the ability to deal with discontinuities while also using a structured mesh, something that it's not possible with cohesive elements since they are only inserted between the bulk elements. However, XFEM also struggles to correctly deal with problems where there are crack intersections.

2.3. Multiscale Methods

Multiscale methods are a group of numerical techniques that combine information from multiple models at the relevant scales. There are many types and formulations, with some being tailored to specific problems

or material groups. In this section, the focus will be on those used to study the behaviour of composites, especially those used to study damage onset and growth and those that analyze this behaviour in 3D.

2.3.1. Classes of multiscale methods

Regarding the transfer of information between the coarse-scale and fine-scale models, multiscale methods can be classified as either hierarchical, concurrent or semiconcurrent [105].

Hierarchical

In hierarchical methods, the fine-scale and coarse-scale models are solved separately, with the transfer of information between scales only occurring before and after the analysis. This class of methods is the most widely used one and is much more computationally efficient than others [105]. The independence of each model is the reason behind this efficiency, but it's also the source of its drawbacks. Because of it, the inputs provided by the coarse-scale model cannot be updated during the analysis, which limits its application to problems where localizing phenomena, such as cracks or shear bands, are present [55]. They are most often used to extract information about the material's behaviour [86, 101].

Concurrent

In concurrent methods, the coarse and fine-scale models are embedded into one another. Since it would require a significant amount of computational effort to apply such embedding on all of the geometry, this coupling between the two models is only employed in specific regions of interest, such as the crack tip/surface in fracture problems. The regions of interest can be identified either through *a priori* knowledge of the problem or during the simulation itself [105]. In order to couple both scales, compatibility and equilibrium equations are enforced at the boundary between both models. Furthermore, depending on the formulation of the method used, a handshake region and/or padding is often required. This region is governed by a mixed formulation and ensures a proper scale transition. Some methods even require further subdivisions of the handshake domain, with different formulations for each [64].

Semiconcurrent

In semiconcurrent multiscale methods, the fine-scale and coarse-scale models are solved simultaneously but are not embedded into each other. Between each model, only specific quantities are transferred, and compatibility and equilibrium equations are only satisfied approximately [105]. The computational cost is similar to that of concurrent methods, since both problems can be treated separately, and the formulation doesn't have to be the same in both scales. The decoupling of the fine and coarse-scale models allows for a more flexible implementation and easier parallelization [89].

One of the most common semiconcurrent methods is the FE^2 method, developed by Feyel and Chaboche [25, 26], which has been extensively used in research. Kouznetsova et al. [52] extended this method for problems involving size effects, such as damage in a CDM (Continuum Damage Mechanics) framework.

2.3.2. Representative Volume Element

For hierarchical and semiconcurrent methods, the microscale problem is modeled by using a Representative Volume Element (RVE), which is a small sample of the material system being studied at a relevant length scale. In order to create an adequate RVE, three aspects must be considered: size, boundary conditions and spacial distribution of phases.

Size

Regarding the size of the RVE, it should be big enough to capture the material behaviour, both in terms of material distribution and characteristic size of the mechanism/feature being studied, but also be as small as possible to minimize the computational cost ([30]).

The existence of an RVE is not guaranteed. The concept of an RVE relies on the separation of scales, which stipulates that the RVE's size, l_m, must be bigger than the size of microscopic fluctuations, l_μ, and much smaller than the macroscopic size l_M [30]. In this context, microscopic fluctuations are the local discrepancies between the micro and macroscale deformation.

$$l_\mu < l_m \ll l_M \tag{2.18}$$

In the case of FRP, three relevant length scales are present: fiber diameter (5-10 μm), ply thickness (100–300 μm) and laminate thickness (above 1 mm) [54]. Furthermore, the existence of an RVE also depends on the

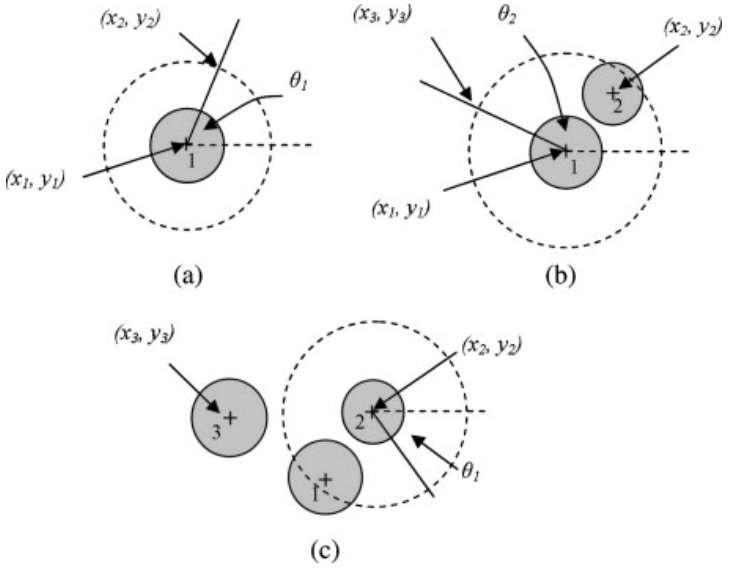


Figure 2.11: Schematic representation of the NNA algorithm: (a) Assignment of fiber's nearest neighbour. (c) Assignment of fiber's second nearest neighbour. (c) Nearest neighbour being assigned for a subsequent fiber. [100]

materials being modelled. An RVE can only be found for statistically homogeneous materials and, even then, in some applications (e.g. softening materials), it might not exist [32].

When an RVE exists, the results obtained from solving its boundary value problem are expected to converge for an RVE size bigger than a critical value, the critical RVE size. In general, analytical estimates for the critical RVE size, such as the one presented by Drugan and Willis [21], are scarce and mostly unreliable, requiring it to be determined numerically by repeating the same analysis multiple times using different RVE sizes. The critical size is also different for each property being studied [53].

For 2D problems involving FRPs, the RVE usually takes the shape of a square, and its size is usually expressed in terms of a parameter $\delta = L/R$, with L being the edge length of the RVE and R the fiber radius. Trias et al. [94] presents an extensive list of constraints on δ in order to obtain a statistically representative RVE. Combining all of them yields $\delta \geq 50$. However, more recent authors have reported converged results for much smaller RVE sizes. Melro et al. [61] had converged results in the estimation of elastic properties for $\delta \geq 30$ and González and LLorca [34] for $\delta \simeq 16$. Furthermore, previous work on the topic of this thesis at TU Delft successfully used 50 μm x 50 μm and 75 μm x 75 μm for R=2.5 μm [92], which corresponds to $\delta = 20$ and $\delta = 30$, respectively.

The conversion of a 2D RVE into a 3D one for problems involving unidirectional fibers is usually done through an extrusion in the third dimension by a given length [54]. The RVE thickness values used tend to be arbitrary [40, 93, 102] for RVEs that model fibers aligned with the out-of-plane direction. The only exception found was the work of Múgica et al. [67], where a thickness of 600 μm was used because it aimed to capture fiber splitting by modifying the elastic properties of the material randomly. For cases where multiple fiber orientations are present in the RVE, Soni et al. [90] used cubic RVEs.

Spatial Distribution of Phases

The spatial distribution of fibers is also of crucial importance when generating RVEs. In order to create a random distribution of non-overlapping fibers within the RVE, a dedicated algorithm must be selected. Some purely numerical examples include those proposed by Buryachenko et al. [5] (Hard-Core Model, HCM and Hard-Core Shaking Model, HCSM), Wang et al. [103] (Initially Periodic Shaking Model, IPSM) and Yang et al. (Random Sequential Expansion, RSE)[107]. Most of these initially developed and purely numeric algorithms struggle to develop statistically relevant RVEs with high fiber volume fractions and for that, more sophisticated algorithms are employed. They might also require significant computational effort or non-trivial algorithms to effectively implement the shaking. Other alternatives are based on experimental data, such as the image-based [42], or a mixture between numerical and experimental, such as the Nearest Neighbour Algorithm (NNA) by Vaughan and McCarthy [100], the latter of which is depicted in Figure 2.11. Besides being fully capable of generating RVEs for almost every fiber volume fraction, they also implicitly include effects in the packing of fibers that are not resultant of geometrical constraints, such as those induced by elastic,

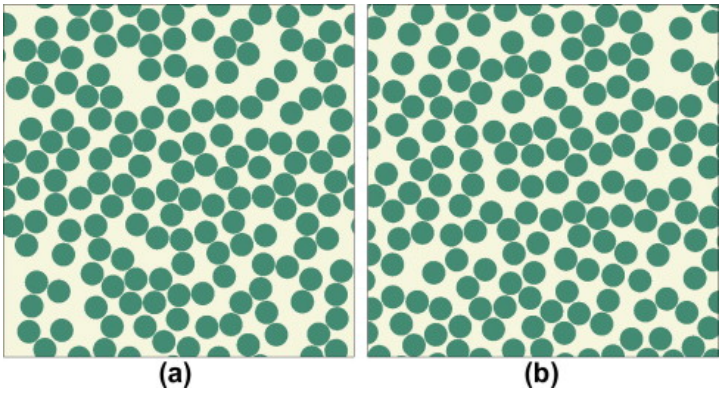


Figure 2.12: Example of two RVEs with (a) and without (b) material periodicity. [61]

hydrodynamic, and cohesive forces [5]. However, these require experimental data which might not be always available.

Another required feature for RVEs when using periodic boundary conditions, which will be discussed later, is material periodicity, and it consists on the phases present on a given edge (2D) or surface (3D) reappearing in the opposite one in the exact same positions, as shown in Figure 2.12. This removes the wall-effect, which makes the material in the RVE behave as if it is somewhere in the middle of a much bigger sample [32]. This is especially important for fracture problems, where a crack propagating in a given phase should reappear on the other side of the RVE in the same phase, because each one is governed by different cohesive laws.

Boundary conditions

The choice of appropriate boundary conditions for the microscopic RVE is also very important and can have a significant influence on the results [71]. There are many ways to prescribe the boundary conditions [30]. Some of them are:

- Voigt-Taylor assumption: no fluctuations within the RVE
- Uniform displacement: no fluctuations only in the RVE boundary
- Periodic boundary conditions (PBCs): the microfluctuation field at a point on the boundary has to be equal to the microfluctuation field at its corresponding point in the opposite edge/face. The traction at the boundary also has to be anti-periodic.
- Sachs-Reuss assumption: constant stress and rotation within the RVE, prescribed by the macroscopic stress tensor

The great majority of the authors employ PBCs [58, 96–98, 106]. Only under specific circumstances, other types of boundary conditions are used, with the usual motivation being simplicity. PBCs are the preferred choice as they provide the most accurate approximation of the material properties and eliminate the wall-effect [54]. Their application in 3D problems is as straightforward as in the 2D case, only with added pairs of surfaces in the extra dimension, having been widely implemented in multiscale problems [37, 62, 67, 87]. The equations for implementing PBCs in a non-redundant way can be found in the work of Varnamkhasti [99].

In the context of fracture, the application of PBCs is not straightforward. The periodicity in the results leads to periodic replicas of the cracks that nucleate within the RVE [102]. Turteltaub et al. [98] avoided this constraint by introducing the concept of an effective macroscopic crack, which is shown to provide results independent of these replicas.

2.3.3. Homogenization

Homogenization procedures are used to translate the behaviour observed at the microscopic scales into an equivalent homogenized macroscale behaviour. The first homogenization methods that were developed were mostly analytical. This allows for an almost effortless scale transition, but is often limited to very simple geometries and makes assumptions about the mechanical response of the RVE, which can then lead to large discrepancies between the obtained results and the actual behaviour. Analytical methods will not be

covered, but a good overview can be found in Geers et al.[30]. Nowadays, computational homogenization procedures are preferred. Since they model and solve the microscopic problem explicitly, no assumption regarding the mechanical response is made, providing very accurate results. It requires, however, an accurate characterization of the individual constituents and respective interfaces in the microstructure, which might not be easy.

Here, first-order homogenization, one of the most widely used computational methods, will be covered in detail. More sophisticated alternatives are briefly mentioned in the end as well.

First-order computational homogenization

First-order homogenization is the simplest computational homogenization scheme. In short, a macroscopic deformation gradient is obtained from the macroscopic problem and transferred to the microscopic boundary value problem, which is then solved (Figure 2.13). From the results, effective or homogenized quantities are extracted and applied to the macroscopic problem which implicitly take into account the heterogeneous microstructure of the problem. It can deal with microstructures of arbitrary morphology and distribution of phases, but it's unable to take into account their absolute size [51]. The mathematical relations that will be presented here are expressed in more detail in the work of Kouznetsova [51] and Geers et al. [29].

Microscopic problem definition

In first-order homogenization, the non-linear macroscopic deformation field is mapped into the microscopic problem by passing the macroscopic deformation gradient $\mathbf{F}_M$, which is then linearized according to the following relation:

$$\Delta\mathbf{x}_m = \mathbf{F}_M \Delta\mathbf{X}_m + \mathbf{w}_m \tag{2.19}$$

where $\mathbf{X}_m$ and $\mathbf{x}_m$ are the position vectors of a point within the RVE in the reference configuration and in the current configuration, respectively. The microfluctuation field, $\mathbf{w}_m$, is the local discrepancy between the micro and macroscale deformation. The microscopic deformation gradient $\mathbf{F}_m$ can then be obtained using

$$\mathbf{F}_m = (\nabla_{0m}\mathbf{x}_m)^T \tag{2.20}$$

As for the equilibrium equation of the RVE, under quasi-static conditions and in the absence of body forces, it reduces to:

$$\mathrm{div}_{0m}(\mathbf{P}_m^T) = \mathbf{0} \tag{2.21}$$

where $\mathbf{P}_m$ is the microscopic first Piola-Kirchhoff stress tensor. With these quantities defined, the microscopic boundary value problem can then be solved.

Micro-to-macro transition relations

Once the microscopic problem has been solved, the obtained results need to be transferred to the macroscopic problem. This is done by obtaining effective quantities for deformation, stress and internal work.

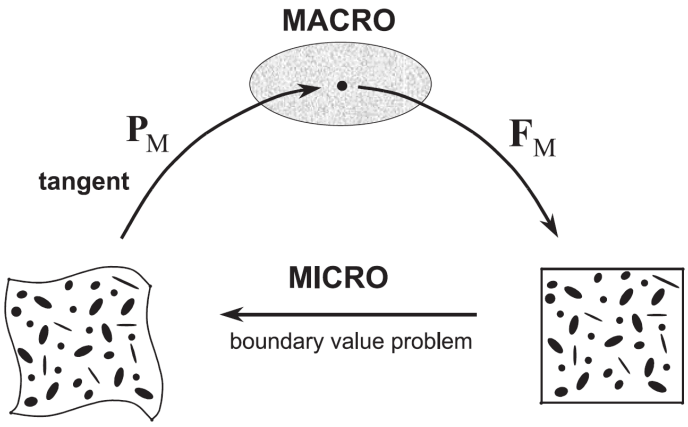


Figure 2.13: Schematic of first-order homogenization. [51]

Deformation

The homogenized macroscopic deformation gradient and the volume averaged Green-Lagrange strain tensor, $\mathbf{E}^*_M$, can be obtained from the microscopic deformation gradient using the following relations

$$\mathbf{F}_M = \frac{1}{V_0}\int_{V_0} \mathbf{F}_m \mathrm{d}V_0 = \frac{1}{V_0}\int_{\partial V_0} \mathbf{x}_m \otimes \mathbf{N}_m \mathrm{d}\partial V_0 \tag{2.22}$$

$$\mathbf{E}^*_M = \frac{1}{2V_0}\int_{V_0} (\mathbf{F}^T_m \mathbf{F}_m - \mathbf{I})\mathrm{d}V_0 \tag{2.23}$$

with ∂V_0 being the boundary of the reference domain and $\mathbf{N}_m$ its outward normal in a given point.

Stress

Likewise, the macroscopic first Piola-Kirchhoff stress tensor is obtained by averaging the microscopic tensor over the volume of the domain. This relation can be expressed in terms of quantities acting on the boundary in the following way

$$\mathbf{P}_M = \frac{1}{V_0}\int_{V_0} \mathbf{P}_m \mathrm{d}V_0 = \frac{1}{V_0}\int_{\partial V_0} \mathbf{s}_m \otimes \mathbf{X}_m \,\mathrm{d}\partial V_0 \tag{2.24}$$

Internal work

Besides enforcing the equivalence between quantities describing deformation and stress, the energy equivalence between the micro and macroscale also needs to be imposed. This is required because, in general, the product of averages isn't equal to the average of products [71, 98]. The Hill-Mandel condition imposes such requirement and can be expressed as

$$\delta W_{0M} = \underbrace{\mathbf{P}_M \delta \mathbf{F}^T_M = \frac{1}{V_0}\int_{V_0} \mathbf{P}_m \delta \mathbf{F}^T_m \mathrm{d}V_0}_{\text{Hill-Mandel's principle}} = \frac{1}{V_0}\int_{\partial V_0} \mathbf{s}_m \cdot \delta \mathbf{x}_m \,\mathrm{d}\partial V_0 \tag{2.25}$$

where $\mathbf{s}_m$ is the traction in the reference configuration.

Alternative homogenization schemes

First-order homogenization is a very versatile technique, allowing the study of the mechanical behaviour of complex materials. Its relatively simple formulation also makes its implementation very straightforward, resulting in a widespread use in most multiscale problems. However, it has some drawbacks that limit its employment in some specific problems. Since it lacks an internal size parameter, first-order homogenization struggles to model problems involving macroscopic localization or microscopic size effects, i.e those where the separation of scales assumption is no longer valid. This fundamental limitation motivated the development of more complex homogenization theories. One of those alternatives is second-order homogenization, in which both the macroscopic deformation gradient and its Lagrangian gradient are transferred to the microscopic problem, as shown in Figure 2.14 [51]. In second-order homogenization, equation 2.19 becomes equation 2.26. This technique is still not as developed as the first-order scheme. It also presents some drawbacks itself, such as its implementation in finite element codes, which is not as straightforward as the first-order version, since the shape functions lack the required higher order continuity [72].

$$\Delta \mathbf{x}_m = \mathbf{F}_M \Delta \mathbf{X}_m + \frac{1}{2}\mathbf{X}_m \nabla_{0M} \mathbf{F}_M \mathbf{X}_m + \mathbf{w}_m \tag{2.26}$$

2.3.4. Composite phases modeling in RVEs

Fibers

Carbon fibers

Individual carbon fibers consist of layers of graphene, which are designated as basal planes. The atomic bonds between atoms within a basal plane are covalent and between planes are of van der Waals nature. Regarding the arrangement of these planes in a fiber, they are aligned with the fiber axis but the arrangement in the transverse direction can vary (Figure 2.15). If the arrangement in this direction forms a pattern, such as

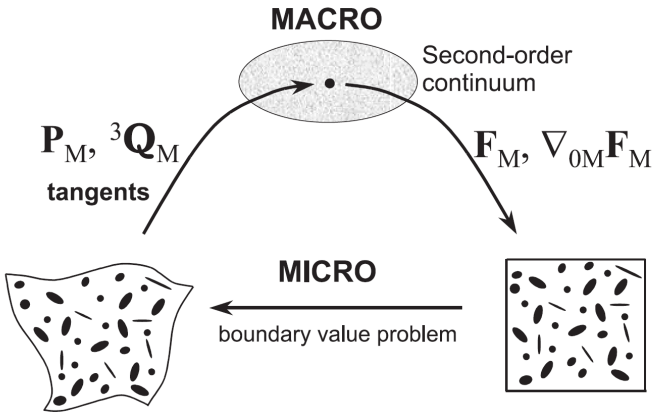


Figure 2.14: Schematic of second-order homogenization. [51]

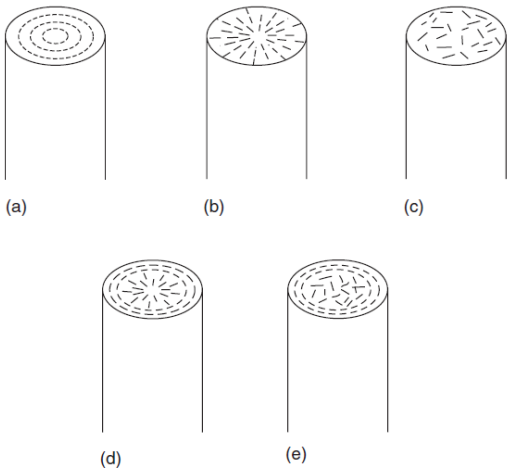


Figure 2.15: Possible arrangements of the basal planes in a fiber: (a) circumferential, (b) radial, (c) random, (d) radial–circumferential, and (e) random–circumferential. [57]

circumferential or radial, the fibers can be modelled as circumferentially orthotropic or radially orthotropic, respectively. These arrangements are, however, very rare and the basal planes are usually arranged in a random way, which leads to a transversely isotropic behaviour [57]. Carbon fibers are consistently modeled as linear elastic transversely isotropic solids in the literature. Since the presence of defects in ceramic materials is also random, scatter in the elastic properties is included in the constitutive model by attributing a random value, following a Weibull distribution, for the elastic properties of the bulk elements in the fibers [37, 69].

Glass fibers

In glass fibers, the atoms (mostly silicon and oxygen) are arranged randomly, since glass has an amorphous structure. This random distribution leads to an equal mechanical behaviour in all directions, making glass fibers behave as linear elastic isotropic solids [54, 90]. When relevant, the variability in the elastic properties can be introduced in the exact same way as it was explained for carbon fibers [67].

Matrix

Thermoset polymers

In thermoset polymers, such as epoxies, the molecules form a cross-linked structure, which is formed during the curing reaction [57]. It is observed that under tension, thermosets display a very brittle behaviour. This behavior is caused by the inability of cross-linked polymers to undergo crazing. However, depending on the temperature and the degree of cure, it can display extensive plastic deformation. Under shear or compression loadings, significant plastic deformation is present, as the polymer fails through shear yielding. Furthermore, the plastic yielding under these circumstances is pressure dependent[50].

In the literature, thermoset polymers are modeled as isotropic elasto-plastic solids. Some authors [54, 101] use the Mohr-Coulomb criteria ([10, 66], summarized in [58]) while others [37, 81, 87, 102] use some modified version of the Drucker-Prager model, which follows the standard model [20] under compression but is controlled by the maximum principal stress criteria for tension [68]. A comparison between the two criteria can be found in the work of Jiang and Xie [46, 47].

Thermoplastic polymers

Thermoplastics, such as PEEK, PEKK or PS, have their molecules arranged in long polymeric chains, which are then entangled and held together through secondary bonds [57]. Regarding the mechanical response, thermoplastic polymers display plastic deformation under all types of loading. The amount of plasticity heavily depends on many variables such as temperature, pressure or the presence of special substances (e.g. crazing agents) in the work environment [50].

In the literature, thermoplastics are modelled as isotropic elasto-plastic solids, in a similar fashion to thermosets, but the original Drucker-Prager model [20] is used as yield criterion instead of the modified version [67].

Interface

The properties of the interface depend on the fiber-matrix system used. Usually, different cohesive laws than those used for the fiber and matrix phases are implemented in the cohesive elements at the interface. Some authors use a simple bi-linear TSL. The parameters that define them can be obtained experimentally through push-in tests [54]. When studying the mechanical behaviour under compression or shear loadings, some authors also include the effect of friction [87].

3

Methodology

In this chapter, the analytical formulation of the multiscale methodology used to extract ETSLs, with the adaptions needed to be employed in 3D problems, is presented.

3.1. Overview

For problems involving localizing phenomena, such as cracks, hierarchical methods are ill-suited. However, previous work at TU Delft [96–98] avoids this constraint by introducing the concept of an effective macroscopic crack, whose response is unaffected by the presence of periodic replicas.

With this methodology, ETSLs governing the response of the effective macroscopic crack can be obtained. These implicitly take into account the microscale response, since they are obtained by homogenizing it. This methodology unlocks the ability to model, in a generic way, complex fracture responses resulting from intricate microstructures. It can also be used to assess the macroscale impact of changes to the microstructure itself, by comparing the changes in the obtained ETSLs when certain characteristics are modified.

In this chapter, the 3D extension of this methodology is presented. This is a relevant step in its evolution, since it removes the plane strain assumption present in 2D. In the context of composite materials, it also enables the study of microsctrutures made up of multiple plies, which may or may not be aligned with the principal directions.

First, the microscale boundary value problem's formulation is given. Then, it is established how quantities in both scales relate to each other at any moment in time. Lastly, the relations used to obtain the quantities that define the effective macroscopic crack in an energetically consistent way are presented.

3.2. Microstructural formulation

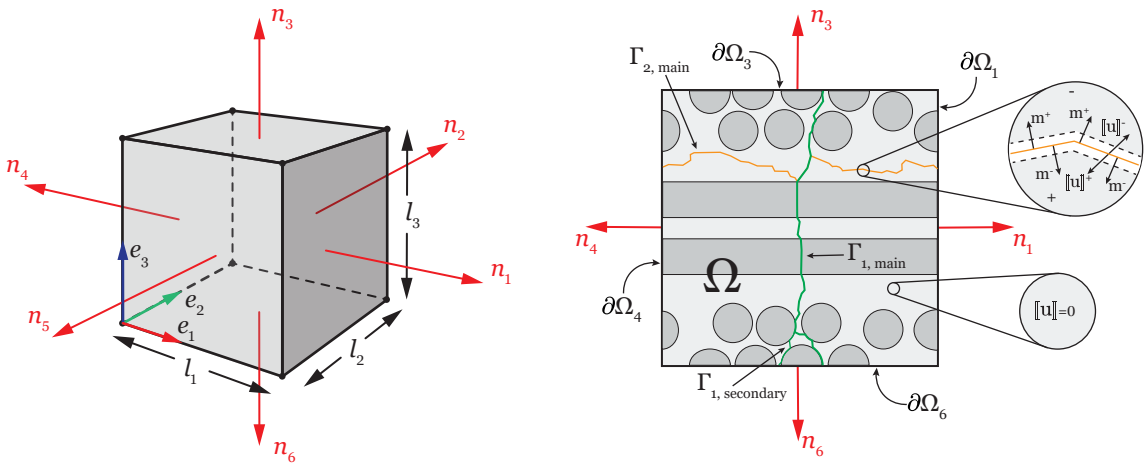


Figure 3.1: Microstructural volume element. Adapted from [92]

In Figure 3.1, the microstructural volume element is schematically shown. In it, Ω denotes the volume element, $\partial\Omega$ its boundary, and Γ a fracture surface. Each face is identified by an index $i = 1,2,...6$ and has an outward pointing normal n_i

The microscopic boundary value problem is formulated as a quasi-static fracture process without body forces present. Traction continuity is also imposed on the fracture surface. This can be expressed, at any time t, as:

$$\text{div}\boldsymbol{\sigma}(\boldsymbol{x},t) = \mathbf{0} \quad \boldsymbol{x} \in \Omega\backslash\Gamma \tag{3.1}$$

$$\boldsymbol{t}^+(\boldsymbol{x}^+,t) = -\boldsymbol{t}^-(\boldsymbol{x}^-,t) \quad \boldsymbol{x} \in \Gamma \tag{3.2}$$

where $\boldsymbol{\sigma}$ is the stress tensor and $\boldsymbol{x}$ the position of a material point w.r.t the reference configuration. The superscripts + and − identify the side of the crack surface from where the quantities are obtained. Furthermore, periodic boundary conditions are also enforced, which can be expressed as:

$$\begin{aligned}
&\boldsymbol{u}(\boldsymbol{x}+l_1\mathbf{e}_1,t) - \boldsymbol{u}(\boldsymbol{x},t) = l_1\overline{\boldsymbol{\varepsilon}}(t)\mathbf{e}_1 \\
&\boldsymbol{t}(\boldsymbol{x}+l_1\mathbf{e}_1,t) = -\boldsymbol{t}(\boldsymbol{x},t) && \boldsymbol{x} \in \partial\Omega_4\backslash\Gamma \\
&\boldsymbol{u}(\boldsymbol{x}+l_2\mathbf{e}_2,t) - \boldsymbol{u}(\boldsymbol{x},t) = l_2\overline{\boldsymbol{\varepsilon}}(t)\mathbf{e}_2 \\
&\boldsymbol{t}(\boldsymbol{x}+l_2\mathbf{e}_2,t) = -\boldsymbol{t}(\boldsymbol{x},t) && \boldsymbol{x} \in \partial\Omega_5\backslash\Gamma \\
&\boldsymbol{u}(\boldsymbol{x}+l_3\mathbf{e}_3,t) - \boldsymbol{u}(\boldsymbol{x},t) = l_3\overline{\boldsymbol{\varepsilon}}(t)\mathbf{e}_3 \\
&\boldsymbol{t}(\boldsymbol{x}+l_3\mathbf{e}_3,t) = -\boldsymbol{t}(\boldsymbol{x},t) && \boldsymbol{x} \in \partial\Omega_6\backslash\Gamma \\
&\boldsymbol{u}^\pm(\boldsymbol{x}^\pm+l_1\mathbf{e}_1,t) - \boldsymbol{u}^\pm(\boldsymbol{x}^\pm,t) = l_1\overline{\boldsymbol{\varepsilon}}(t)\mathbf{e}_1 \\
&\boldsymbol{t}^\pm(\boldsymbol{x}^\pm+l_1\mathbf{e}_1,t) = -\boldsymbol{t}^\pm(\boldsymbol{x}^\pm,t) && \boldsymbol{x} \in \partial\Omega_4\cap\Gamma \\
&\boldsymbol{u}^\pm(\boldsymbol{x}^\pm+l_2\mathbf{e}_2,t) - \boldsymbol{u}^\pm(\boldsymbol{x}^\pm,t) = l_2\overline{\boldsymbol{\varepsilon}}(t)\mathbf{e}_2 \\
&\boldsymbol{t}^\pm(\boldsymbol{x}^\pm+l_2\mathbf{e}_2,t) = -\boldsymbol{t}^\pm(\boldsymbol{x}^\pm,t) && \boldsymbol{x} \in \partial\Omega_5\cap\Gamma \\
&\boldsymbol{u}^\pm(\boldsymbol{x}^\pm+l_3\mathbf{e}_3,t) - \boldsymbol{u}^\pm(\boldsymbol{x}^\pm,t) = l_3\overline{\boldsymbol{\varepsilon}}(t)\mathbf{e}_3 \\
&\boldsymbol{t}^\pm(\boldsymbol{x}^\pm+l_3\mathbf{e}_3,t) = -\boldsymbol{t}^\pm(\boldsymbol{x}^\pm,t) && \boldsymbol{x} \in \partial\Omega_6\cap\Gamma
\end{aligned} \tag{3.3}$$

where $\overline{\boldsymbol{\varepsilon}}$ is the prescribed macroscopic strain tensor.

For all points that are not in the crack surface, the microscale strain field, $\boldsymbol{\varepsilon}$, is obtained from the displacement field,u, through the kinematic relation for small strains, which can be written as:

$$\boldsymbol{\varepsilon} = \tfrac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}}\right) \quad \boldsymbol{x} \in \Omega\backslash\Gamma \tag{3.4}$$

where ∇ denotes the gradient operator and the superscript T the transpose.

In terms of constitutive relations, all constituents are assumed to behave as linear elastic materials until fracture. Once formed, the crack surface's constitutive response is governed by a traction-separation relation. This can be expressed as:

$$\boldsymbol{\sigma} = \mathbb{C}\boldsymbol{\varepsilon} \quad \boldsymbol{x} \in \Omega\backslash\Gamma \tag{3.5}$$

$$\boldsymbol{t} = \mathbf{f}_{\text{coh}}\left(\llbracket\mathbf{u}\rrbracket, \boldsymbol{\kappa}, \boldsymbol{m}\right) \quad \boldsymbol{x} \in \Gamma \tag{3.6}$$

where $\mathbb{C}$ is the stiffness tensor. For each phase and interface, a different traction-separation relation, $\mathbf{f}_{\text{coh}}$, is used, which is a function of the crack normal $\boldsymbol{m}$, the damage variable(s) $\boldsymbol{\kappa}$ and the crack opening $\llbracket\mathbf{u}\rrbracket$. The latter is defined as:

$$\llbracket\mathbf{u}\rrbracket = \mathbf{u}^+ - \mathbf{u}^- \quad \boldsymbol{x} \in \Gamma \tag{3.7}$$

which, when combined with the equations for periodic boundary conditions, yields the following relations:

$$\begin{aligned}
\llbracket\mathbf{u}(\boldsymbol{x},t)\rrbracket &= \llbracket\mathbf{u}(\boldsymbol{x}+l_1\mathbf{e}_1,t)\rrbracket \quad \boldsymbol{x} \in \partial\Omega_4\cap\Gamma \\
\llbracket\mathbf{u}(\boldsymbol{x},t)\rrbracket &= \llbracket\mathbf{u}(\boldsymbol{x}+l_2\mathbf{e}_2,t)\rrbracket \quad \boldsymbol{x} \in \partial\Omega_5\cap\Gamma \\
\llbracket\mathbf{u}(\boldsymbol{x},t)\rrbracket &= \llbracket\mathbf{u}(\boldsymbol{x}+l_3\mathbf{e}_3,t)\rrbracket \quad \boldsymbol{x} \in \partial\Omega_6\cap\Gamma
\end{aligned} \tag{3.8}$$

3.3. Micro/macro scale coupling

Having formulated the microstructural problem, the way quantities are translated between scales will now be described.

3.3.1. Strain relations

As mentioned before, the microscale problem is loaded by prescribing the macroscopic strain tensor. This tensor is defined as:

$$\overline{\boldsymbol{\varepsilon}} = \frac{1}{|\Omega|}\int_{\partial\Omega} [[\mathbf{u}]] \otimes \boldsymbol{n}]_{\text{sym}}\, \mathrm{d}s \tag{3.9}$$

where the subscript "sym" indicates the symmetric part of the tensor. As demonstrated in Turteltaub et al. [98], in the presence of a crack, the applied strain can be decomposed at any time instant in the following way:

$$\overline{\boldsymbol{\varepsilon}} = \boldsymbol{\varepsilon}^{\text{b}} + \boldsymbol{\varepsilon}^{\text{f}} \tag{3.10}$$

where $\boldsymbol{\varepsilon}^{\text{b}}$ is the bulk strain and $\boldsymbol{\varepsilon}^{\text{f}}$ is the fracture strain, which in turn are respectively given by:

$$\boldsymbol{\varepsilon}^{\text{b}} = \langle \boldsymbol{\varepsilon} \rangle_{\Omega} = \frac{1}{|\Omega|}\int_{\Omega} \boldsymbol{\varepsilon}\, \mathrm{d}v \tag{3.11}$$

$$\boldsymbol{\varepsilon}^{\text{f}} = \frac{1}{|\Omega|}\int_{\Gamma} [[\mathbf{u}]] \otimes \boldsymbol{m}]_{\text{sym}}\, \mathrm{d}s \tag{3.12}$$

where the notation $\langle \cdot \rangle_{\Omega} = (1/|\Omega|)\int_{\Omega}(\cdot)\,\mathrm{d}v$ is used to represent averaged quantities.

3.3.2. Stress relations

Likewise, the externally applied (macroscopic) stress tensor is defined as:

$$\overline{\boldsymbol{\sigma}} = \frac{1}{|\Omega|}\int_{\partial\Omega} \boldsymbol{t} \otimes \boldsymbol{x}\, \mathrm{d}s \tag{3.13}$$

Once again, as demonstrated in Turteltaub et al.[98], the applied (macroscopic) stress tensor matches the volume-averaged stress tensor i.e:

$$\overline{\boldsymbol{\sigma}} = \langle \boldsymbol{\sigma} \rangle_{\Omega} \tag{3.14}$$

with $\langle \boldsymbol{\sigma} \rangle_{\Omega}$ being defined as:

$$\langle \boldsymbol{\sigma} \rangle_{\Omega} = \frac{1}{|\Omega|}\int_{\Omega} \boldsymbol{\sigma}\, \mathrm{d}v \tag{3.15}$$

3.3.3. Power relations

The externally applied power density (per unit volume) done on the boundary of the volume element is defined as:

$$P^{\text{ext}} = \frac{1}{|\Omega|}\int_{\partial\Omega} \boldsymbol{t} \cdot \dot{\mathbf{u}}\, \mathrm{d}s \tag{3.16}$$

From this quantity, a fracture power density, P^{f}, and a bulk power density, P^{b}, can be extracted, which are given by:

$$P^{\text{ext}} = P^{\text{b}} + P^{\text{f}} \tag{3.17}$$

$$P^{\text{b}} = \langle P \rangle_{\Omega} = \frac{1}{|\Omega|}\int_{\Omega} \boldsymbol{\sigma} \cdot \dot{\boldsymbol{\varepsilon}}\, \mathrm{d}v \tag{3.18}$$

$$P^{\text{f}} = \frac{1}{|\Omega|}\int_{\Gamma} \boldsymbol{t} \cdot [[\dot{\mathbf{u}}]]\, \mathrm{d}s \tag{3.19}$$

On the other hand, the macroscopic power is defined as:

$$P^{\text{M}} = \boldsymbol{\sigma}^{\text{M}} \cdot \dot{\boldsymbol{\varepsilon}}^{\text{M}} = \langle \boldsymbol{\sigma} \rangle_{\Omega} \cdot \dot{\overline{\boldsymbol{\varepsilon}}} \tag{3.20}$$

3.3.4. Hill-Mandel condition

The Hill-Mandel condition states that for a homogenization procedure to be valid, the scale transition needs to be energetically consistent, i.e the macroscopic and microscopic power densities must match:

$$P^{\text{M}} = P^{\text{ext}} \tag{3.21}$$

This global form of the Hill-Mandel condition (since it takes both bulk and fracture contributions into account) is satisfied *a priori* when PBCs are used, even in the presence of a crack. However, when a localized

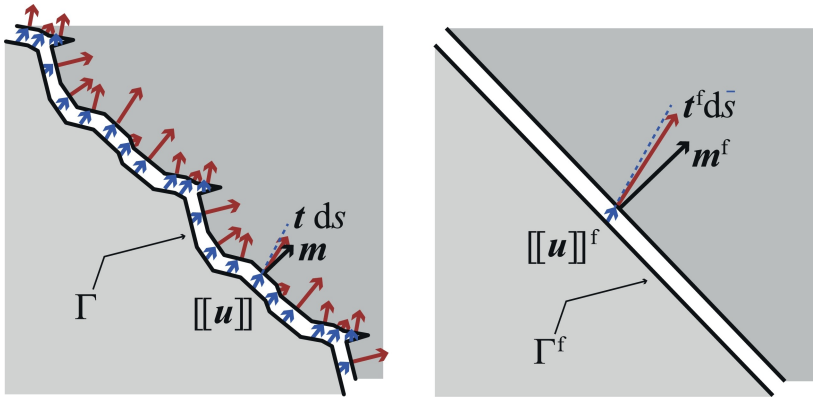


Figure 3.2: Comparison between microscale and macroscale crack representation. All quantities that define the macroscale crack are obtained directly from their microscale equivalents. Both of these cracks have the same rate of work, due to the crack-based Hill-Mandel condition that couples them. Obtained from [98].

crack forms, having the volume element converge to an RVE is challenging, as the ratio between the crack surface's area and the volume element's volume changes while increasing the volume element's size. As a result, the crack's response has to be isolated from that of the surrounding material and used in a separate scale transition.

In this crack-based scale transition, the microscopic crack domain is homogenized into a straight equivalent macroscopic crack (see Fig. 3.2), which follows a traction-separation response that encapsulates all relevant information from the microscale. This response is defined in the following way:

$$\boldsymbol{t}^{\mathrm{f}} = \mathbf{f}^{\mathrm{f}}_{\mathrm{coh}}\left(\llbracket \mathbf{u} \rrbracket^{\mathrm{f}}, \boldsymbol{\kappa}^{\mathrm{f}}, \boldsymbol{m}^{\mathrm{f}}\right) \tag{3.22}$$

which depends on the same parameters used in equation 3.6 and where the subscript f denotes effective quantities. For this scale transition, a dedicated crack-based Hill-Mandel condition needs to be verified as well, and its given by:

$$|\Omega| P^{\mathrm{f}}_{\Gamma} = \underbrace{\int_{\Gamma} \boldsymbol{t} \cdot \llbracket \dot{\mathbf{u}} \rrbracket \, \mathrm{d}s}_{\text{Microscopic}} = \underbrace{|\Gamma|^{\mathrm{f}} \boldsymbol{t}^{\mathrm{f}} \cdot \llbracket \dot{\mathbf{u}} \rrbracket^{\mathrm{f}}}_{\text{Macroscopic}} \tag{3.23}$$

where $|\Gamma|^{\mathrm{f}}$ is the effective (macroscopic) crack surface area. Unlike its global counterpart, the crack-based Hill-Mandel condition isn't necessarily satisfied by simply applying PBCs. Furthermore, since in general, the product of averages isn't equal to the average of products, the effective traction and effective crack opening cannot be simultaneously defined as averages of their microscale equivalents.

3.4. Effective quantities

So far, the reasoning as to why part of the response concerning the cracked domain is treated differently and the resulting equivalent macroscopic crack(s), it's now time to present how the effective quantities that describe a macroscopic crack are obtained from the microscopic crack domain.

First, these quantities are obtained from the periodic crack(s) within the crack domain, which also contains branches and isolated segments. A detailed description of what a periodic crack is and how it is identified can be found in Turteltaub et al.[98]. Unless a periodic crack has formed, the volume element is only considered as being partially cracked since load can still be carried through it. Multiple periodic cracks might be present in a fully-cracked volume element due to anisotropy and each of them has a separate traction-separation response.

For a given periodic crack, the effective crack orientation is obtained by averaging the crack normals of all segments that form the periodic crack:

$$\boldsymbol{m}^{\mathrm{f}} = \langle \boldsymbol{m} \rangle_{\Gamma} = \frac{1}{|\Gamma|} \int_{\Gamma} \boldsymbol{m} \, \mathrm{d}s \tag{3.24}$$

with $|\Gamma|$ denoting the area of all segments that form the periodic crack. As for the effective crack surface area, it is obtained by adding up the area of all those same segments projected onto a plane normal to the effective crack orientation:

$$|\Gamma^{\mathrm{f}}| = \int_{\Gamma} \boldsymbol{m} \cdot \boldsymbol{m}^{\mathrm{f}} \, \mathrm{d}s \tag{3.25}$$

Alternatively, this quantity can be obtained from the effective crack normal and from the volume element's size in the following way:

$$|\Gamma^{\mathrm{f}}| = \begin{cases} |\Gamma^{\mathrm{f}}_{\mathrm{min}}| & \text{if} \quad r \geq r_{\mathrm{max}} \\ |\Gamma^{\mathrm{f}}_{\mathrm{max}}| & \text{if} \quad r < r_{\mathrm{max}} \end{cases} \tag{3.26}$$

where:

$$|\Gamma^{\mathrm{f}}_{\mathrm{min}}| = \min\left(\frac{l_1 l_2}{|\boldsymbol{n}_3 \cdot \boldsymbol{m}^{\mathrm{f}}|}, \frac{l_1 l_3}{|\boldsymbol{n}_2 \cdot \boldsymbol{m}^{\mathrm{f}}|}, \frac{l_2 l_3}{|\boldsymbol{n}_1 \cdot \boldsymbol{m}^{\mathrm{f}}|} \right) \tag{3.27}$$

$$|\Gamma^{\mathrm{f}}_{\mathrm{max}}| = \max\left(\frac{l_1 l_2}{|\boldsymbol{n}_3 \cdot \boldsymbol{m}^{\mathrm{f}}|}, \frac{l_1 l_3}{|\boldsymbol{n}_2 \cdot \boldsymbol{m}^{\mathrm{f}}|}, \frac{l_2 l_3}{|\boldsymbol{n}_1 \cdot \boldsymbol{m}^{\mathrm{f}}|} \right) \tag{3.28}$$

$$r = \frac{|\Gamma^{\mathrm{f}}_{\mathrm{max}}|}{|\Gamma^{\mathrm{f}}_{\mathrm{min}}|} \tag{3.29}$$

and r_{max} is a prescribed cut-off value to determine when the effective crack normal is sufficiently close to one of the principal directions, thus limiting the number of times the crack surface is allowed to reenter the volume element.

With these two quantities, the geometry of the respective equivalent macroscopic crack is established. In order to define its response, the effective traction, $\boldsymbol{t}^{\mathrm{f}}$, and effective crack opening, $[\![\dot{\mathbf{u}}]\!]^{\mathrm{f}}$, need to be obtained. As mentioned before, for consistency reasons, these quantities cannot be determined simultaneously by averaging their values over the crack. In order to deal with this issue, a methodology was initially proposed [98]. In it, the effective crack opening is obtained directly through crack averaging. On the other hand, the effective traction is obtained through a weighted average between two effective traction values (one from the crack averaging and the other from the bulk stress and effective crack orientation), which is adjusted using an α parameter such that Hill-Mandel's condition is satisfied, approximately, as best as possible. Since then, two new methods by Turteltaub and R. Suárez-Millán [97] have been developed. For them, one of the quantities (effective traction or effective crack opening) is obtained through averaging, and the other is obtained directly from Hill-Mandel's condition. With these methods, Hill-Mandel's condition is satisfied exactly. Out of these two possibilities, the method where the effective traction is obtained through crack averaging and the effective crack opening is obtained from Hill-Mandel's condition was found to be the most robust option. This method is the one presented here.

As mentioned, the effective traction is obtained by averaging the traction acting on all segments that form the periodic crack:

$$\boldsymbol{t}^{\mathrm{f}} = \boldsymbol{t}^{\mathrm{f}}_{\Gamma} = \frac{1}{|\Gamma|} \int_{\Gamma} \boldsymbol{t} \, \mathrm{d}s \tag{3.30}$$

For the effective crack opening, the effective crack opening rate that satisfies Hill-Mandel's condition can be computed as follows:

$$[\![\dot{\mathbf{u}}]\!]^{\mathrm{f}} = [\![\dot{\mathbf{u}}]\!]^{\mathrm{f,HM}}_{\Gamma} = \beta [\![\dot{\mathbf{u}}]\!]^{\mathrm{f}}_{\Gamma} \tag{3.31}$$

$$\beta = \frac{\displaystyle\int_{\Gamma} \boldsymbol{t} \cdot [\![\dot{\mathbf{u}}]\!] \, \mathrm{d}s}{|\Gamma^{\mathrm{f}}| \boldsymbol{t}^{\mathrm{f}}_{\Gamma} \cdot [\![\dot{\mathbf{u}}]\!]^{\mathrm{f}}_{\Gamma}} \tag{3.32}$$

where $[\![\dot{\mathbf{u}}]\!]^{\mathrm{f}}_{\Gamma}$ is the crack-averaged effective opening rate, given by:

$$[\![\dot{\mathbf{u}}]\!]^{\mathrm{f}}_{\Gamma} = \frac{1}{|\Gamma^{\mathrm{f}}|} \int_{\Gamma} [\![\dot{\mathbf{u}}]\!] \left(\boldsymbol{m} \cdot \boldsymbol{m}^{\mathrm{f}} \right) \mathrm{d}s \tag{3.33}$$

From the effective crack opening rate, the effective crack opening, $[\![\mathbf{u}]\!]^{\mathrm{f}}_{\Gamma}$, is obtained through time integration.

4

Numerical Implementation: Pre-processing

This chapter touches upon the way the numerical models used in the analyses are created. It is divided in two sections. In section §4.1, the data that describes the final model is presented and in section §4.2 the architecture and algorithms of the software used to create the models in an automatic way are described.

4.1. Finite Element Model

The finite element model, created in Abaqus, consists of some fibers within a box volume (Fig. 4.1). The fibers are organized in plies and for each of these, a random fiber distribution is used. Periodic boundary conditions are used, and as a result, the geometry is also periodic at the boundaries. Cohesive elements are inserted between all bulk elements to model the fracture behaviour. All data required to replicate the model will be presented next.

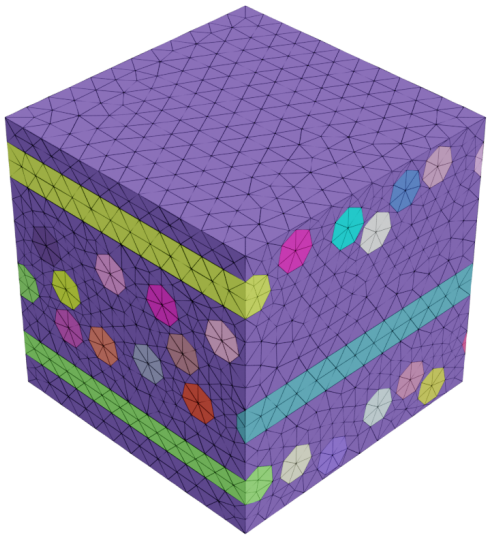


Figure 4.1: Example of a meshed geometry.

4.1.1. Elements

Any mesh created for these analyses uses two different types of elements. For the bulk part, C3D4 elements (4-node linear tetrahedrons) are used. As described in section §2.2.2, fracture can be modelled by inserting cohesive elements between bulk elements. Here, COH3D6 elements (6-node three-dimensional cohesive elements) are employed. The numbering and position of nodes and integration points, in both elements is shown in Figure 4.2. All cohesive elements added have an initial geometrical thickness of zero (i.e faces 1-2-3 and 4-5-6 are coincident at the start of the analysis) and the constitutive thickness is kept at its default value of 1, thus making the strain and opening values equal.

These elements were chosen simply because they are the 3D equivalents of the elements used in the previous implementation (CPE3 and COH2D4) [92].

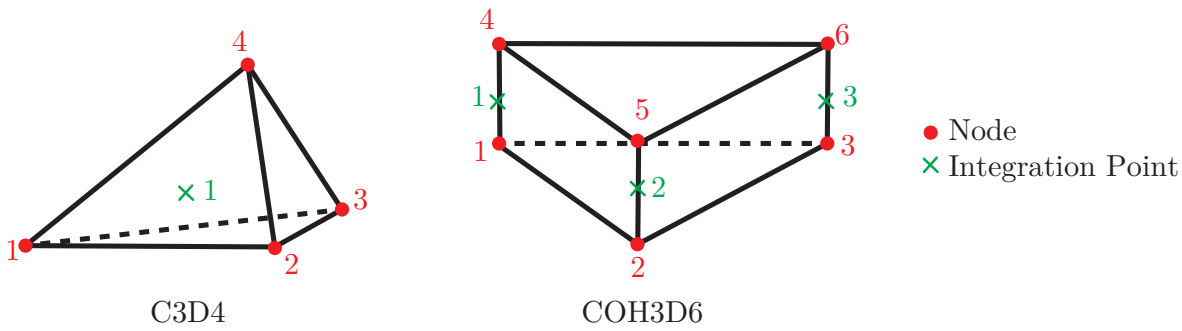


Figure 4.2: Position and numbering scheme of nodes and integration points in elements C3D4 and COH3D6. Adapted from [14, 15]

4.1.2. Materials

Having picked the elements, in order to fully describe their mechanical behaviour, material properties are required. For all the analyses performed, the material properties used in the 2D analyses [92] were not changed for the 3D analyses. They are based on HexTow IM7 carbon fibers and HexPly 8552 epoxy matrix. Such properties are also kept constant throughout the model and no damage is present at the start.

Bulk

For the bulk part of the mesh and in accordance with what was seen in section §2.3.4, the fiber's material is modelled as being a linear elastic transversely isotropic homogeneous solid, using the properties given in Table 4.1. On the other hand, the matrix material is modelled as a linear elastic isotropic homogeneous solid, using the properties given in Table 4.2.

Table 4.1: Material properties used in fiber phase.

Property	Value	Unit
ρ_f	1.78E-06	ton/mm^3
E_{1f}	275E+03	MPa
E_{2f}	20E+03	MPa
E_{3f}	20E+03	MPa
ν_{12f}	0.2	-
ν_{13f}	0.2	-
ν_{23f}	0.333	-
G_{12f}	27.5E+03	MPa
G_{13f}	27.5E+03	MPa
G_{23f}	7.5E+03	MPa
Note: Direction 1 is the fiber's axial direction		

Table 4.2: Material properties used in matrix phase.

Property	Value	Unit
ρ_m	1.31E-06	ton/mm^3
E_m	4E+03	MPa
ν_m	0.35	-

Cohesive

The cohesive element's constitutive response is defined by a bilinear intrinsic TSL (see Fig. 2.6e) for each opening mode. Since a mode-independent fracture behaviour is assumed, the TSLs used for each mode are identical. Such response can be separated in three stages: initial response, damage initiation and damage evolution. All expressions presented here are obtained from the Abaqus documentation [12].

The initial response is linear elastic, where a stiffness matrix, **E**, relates all traction, **t**, and opening, $\boldsymbol{\delta}$, components. This can be written as:

$$\mathbf{t} = \begin{bmatrix} t_n \\ t_s \\ t_t \end{bmatrix} = \begin{bmatrix} E_{nn} & E_{ns} & E_{nt} \\ E_{ns} & E_{ss} & E_{st} \\ E_{nt} & E_{st} & E_{tt} \end{bmatrix} \begin{bmatrix} \delta_n \\ \delta_s \\ \delta_t \end{bmatrix} = \begin{bmatrix} E_{\text{coh}} & 0 & 0 \\ 0 & E_{\text{coh}} & 0 \\ 0 & 0 & E_{\text{coh}} \end{bmatrix} \begin{bmatrix} \delta_n \\ \delta_s \\ \delta_t \end{bmatrix} = \mathbf{E}\boldsymbol{\delta} \tag{4.1}$$

where the subscripts n, s and t denote the normal and two tangential components, respectively. For this problem, it was assumed that there's no coupling between the traction and separation components of different directions, making all non-diagonal components equal to zero. Additionally, since mode-independent fracture is also assumed, all the diagonal stiffness matrix values are equal and denoted by E_{coh}. A high value is used for E_{coh} to minimize the artificial compliance, but not excessively high to avoid numerical errors due to ill-conditioned matrices.

The damage onset occurs when the damage initiation criterion is met. For this problem, the Maximum Nominal Stress criterion was used, which can be expressed as:

$$\max\left\{\frac{\langle t_n \rangle}{t_n^o}, \frac{t_s}{t_s^o}, \frac{t_t}{t_t^o}\right\} = 1 \tag{4.2}$$

with t_n^o,t_s^o and t_t^o being the ultimate traction values in each opening direction and the operator $\langle\cdot\rangle$ being the Macaulay's operator ($\langle\cdot\rangle$=max($\cdot$,0)).

Once damage forms, the damage evolution is controlled by a scalar variable $D \in [0,1]$, which evolves according to the specified criteria

$$t_n = \begin{cases} (1-D)\bar{t}_n & \text{for} \quad \bar{t}_n \geq 0 \\ \bar{t}_n & \text{for} \quad \bar{t}_n < 0 \end{cases} \tag{4.3}$$

$$t_s = (1-D)\bar{t}_s \tag{4.4}$$

$$t_t = (1-D)\bar{t}_t \tag{4.5}$$

$$\tag{4.6}$$

where $\bar{t}_i = E_{\text{coh}}\delta_i$. For this problem, linear damage evolution based on energy is considered, which means that the damage variable evolves according to:

$$\mathrm{D} = \frac{\delta_m^f\left(\delta_m^{\max} - \delta_m^o\right)}{\delta_m^{\max}\left(\delta_m^f - \delta_m^o\right)} \tag{4.7}$$

$$\delta_m^f = \frac{2G^f}{T_{\text{eff}}^o} \tag{4.8}$$

with $\delta_m^{\max}$ being the maximum effective displacement observed during the loading history (see effective displacement definition in eq. 2.14), δ_m^o the effective displacement at damage initiation, G^f the fracture energy and T_{eff}^o the effective traction at damage initiation.

Table 4.3: Material properties used for cohesive phases.

Interface Type	E_{coh} [MPa]	t^{o} t_n [MPa]	t_s [MPa]	t_t [MPa]	G_f [kJ/mm^3]
Fiber - Fiber	1E+09	5000	5000	5000	7.5
Fiber - Matrix	1E+08	85	85	85	0.2
Matrix - Matrix	1E+08	80	80	80	0.2

The cohesive model parameters used can be found in Table 4.3. Once again, these are the same values used in the 2D simulations performed by Suárez-Millán [92]. They were not changed to facilitate the comparison of the results with what was observed previously and because they already provided physically sound results in 2D.

4.1.3. Boundary Conditions

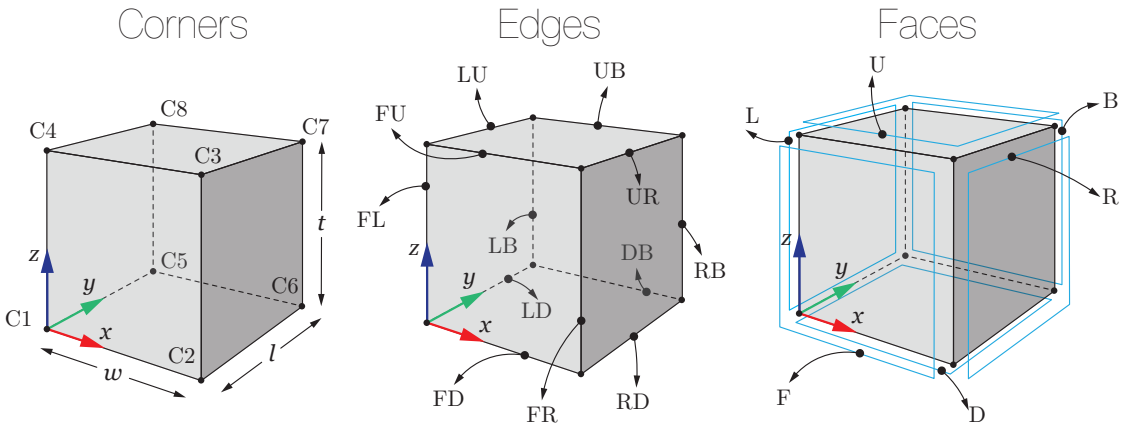


Figure 4.3: Nomenclature used for corners, edges and faces.

As mentioned earlier, multiscale problems of this nature use PBCs in the majority of the cases, since they provide the most accurate approximation of the material properties and eliminate the wall-effect.

Before introducing the equations that govern PBCs and describing how these are implemented in the Abaqus model, the nomenclature used to describe the faces, edges and corners (Fig. 4.3) is explained. The coordinate system used has the Z axis aligned with the stacking direction and the Y axis aligned with the 0° fiber direction. The 6 faces are all named in a way such that they have a unique letter identifier. These are: **D**own, **U**p, **L**eft, **R**ight, **F**ront and **B**ack. An easy way to keep track of this naming convention is by imagining that the 2D analyses were performed on the Front face, with all the other names being given according to this perspective. As for the edges, their names are derived from which face intersection originates them. For example, edge FL is formed by intersecting the Front and Left faces. Lastly, all corners have a number which is obtained by keeping the numbers used in the 2D implementation on the Front face and then replicating the numbering order in the Back face.

The non-redundant set of equations for implementing PBCs in 3D problems can be found in the work of Varnamkhasti [99] and are, for the previously mentioned nomenclature, as follows:

Faces

$$\begin{aligned} \mathbf{u}_{\mathrm{U}} - \mathbf{u}_{\mathrm{D}} - \mathbf{u}^{\mathrm{DU}}_{\mathrm{DUMMY}} &= 0 \\ \mathbf{u}_{\mathrm{R}} - \mathbf{u}_{\mathrm{L}} - \mathbf{u}^{\mathrm{LR}}_{\mathrm{DUMMY}} &= 0 \\ \mathbf{u}_{\mathrm{B}} - \mathbf{u}_{\mathrm{F}} - \mathbf{u}^{\mathrm{FB}}_{\mathrm{DUMMY}} &= 0 \end{aligned} \tag{4.9}$$

Edges

Edges aligned with the X direction

$$\begin{aligned} &\mathbf{u}_{\mathrm{FD}} \rightarrow \text{Independent} \\ &\mathbf{u}_{\mathrm{FU}} - \mathbf{u}_{\mathrm{FD}} - \mathbf{u}^{\mathrm{DU}}_{\mathrm{DUMMY}} = 0 \\ &\mathbf{u}_{\mathrm{DB}} - \mathbf{u}_{\mathrm{FD}} - \mathbf{u}^{\mathrm{FB}}_{\mathrm{DUMMY}} = 0 \\ &\mathbf{u}_{\mathrm{UB}} - \mathbf{u}_{\mathrm{FD}} - \mathbf{u}^{\mathrm{FB}}_{\mathrm{DUMMY}} - \mathbf{u}^{\mathrm{DU}}_{\mathrm{DUMMY}} = 0 \end{aligned} \tag{4.10}$$

Edges aligned with the Y direction

$$\begin{aligned}
&\mathbf{u}_{LD} \rightarrow \text{Independent} \\
&\mathbf{u}_{LU} - \mathbf{u}_{LD} - \mathbf{u}^{DU}_{DUMMY} = 0 \\
&\mathbf{u}_{RD} - \mathbf{u}_{LD} - \mathbf{u}^{LR}_{DUMMY} = 0 \\
&\mathbf{u}_{UR} - \mathbf{u}_{LD} - \mathbf{u}^{LR}_{DUMMY} - \mathbf{u}^{DU}_{DUMMY} = 0
\end{aligned} \tag{4.11}$$

Edges aligned with the Z direction

$$\begin{aligned}
&\mathbf{u}_{FL} \rightarrow \text{Independent} \\
&\mathbf{u}_{FR} - \mathbf{u}_{FL} - \mathbf{u}^{LR}_{DUMMY} = 0 \\
&\mathbf{u}_{LB} - \mathbf{u}_{FL} - \mathbf{u}^{FB}_{DUMMY} = 0 \\
&\mathbf{u}_{RB} - \mathbf{u}_{FL} - \mathbf{u}^{FB}_{DUMMY} - \mathbf{u}^{LR}_{DUMMY} = 0
\end{aligned} \tag{4.12}$$

Corners

$$\begin{aligned}
&\mathbf{u}_{C1} \rightarrow \text{Independent} \\
&\mathbf{u}_{C2} - \mathbf{u}_{C1} - \mathbf{u}^{LR}_{DUMMY} = 0 \\
&\mathbf{u}_{C3} - \mathbf{u}_{C1} - \mathbf{u}^{LR}_{DUMMY} - \mathbf{u}^{DU}_{DUMMY} = 0 \\
&\mathbf{u}_{C4} - \mathbf{u}_{C1} - \mathbf{u}^{DU}_{DUMMY} = 0 \\
&\mathbf{u}_{C5} - \mathbf{u}_{C1} - \mathbf{u}^{FB}_{DUMMY} = 0 \\
&\mathbf{u}_{C6} - \mathbf{u}_{C1} - \mathbf{u}^{LR}_{DUMMY} - \mathbf{u}^{FB}_{DUMMY} = 0 \\
&\mathbf{u}_{C7} - \mathbf{u}_{C1} - \mathbf{u}^{LR}_{DUMMY} - \mathbf{u}^{FB}_{DUMMY} - \mathbf{u}^{DU}_{DUMMY} = 0 \\
&\mathbf{u}_{C8} - \mathbf{u}_{C1} - \mathbf{u}^{FB}_{DUMMY} - \mathbf{u}^{DU}_{DUMMY} = 0
\end{aligned} \tag{4.13}$$

Besides all the constraints added to employ PBCs, the translational DOFs of corner C1 are also fixed to prevent rigid-body motion due to numerical errors.

In the set of equations for PBCs, the $\mathbf{u}_{DUMMY}$ terms refer to the nodal displacements of dummy nodes. These equations are applied using Abaqus' equations, which are actually linear MPCs (Multi-Point Constraints) [16]. Their implementation requires equations to be specified in the format $A_i u_i + A_j u_j + ... = 0$. Because of this, the introduction of terms that don't depend on a nodal displacement is not allowed and dummy nodes are created to circumvent this. The loading is directly applied to them as a prescribed displacement (see section §4.1.4) and their nodal displacements are then used in the equations as shown previously. In total, 3 dummy nodes are used (one for each direction).

Since all equations relate two nodes (excluding dummy nodes), the mesh used needs to be periodic to ensure that every node has a pair. However, in order to add cohesive elements, the nodes of the original mesh must be replicated, which complicates the identification of node pairs as there are multiple coincident nodes. In order for PBCs to be applied correctly, only boundary nodes must be coupled. A boundary node is defined as a node that belongs to a boundary face, which in turn is defined as a set of 3 nodes from the same element that all lie on the RVE's boundary. In Figure 4.4, the correct way of forming pairs using boundary nodes (red dots) is shown. If the node pairing is not done this way, the cohesive element's opening is restricted.

Different types of equations have been presented: for faces, edges and corners. The type of equation used is based on how many boundary faces (of different RVE faces) share the same node. Face, edge and corner equations are applied to nodes shared by one, two and three boundary faces, respectively, after the nodes are replicated to allow the introduction of cohesive elements, as shown in Figure 4.5. The decision of which type is implemented is not solely based on the node's location due to the presence of cohesive elements. For two boundary faces that meet at an edge or corner, if they are split by a cohesive element, the type of equation used for the shared nodes (before replication) is "downranked". Taking the mesh of Figure 4.5 as an example, both the two orange faces and the purple and grey faces make pairs of faces with shared nodes before cohesive

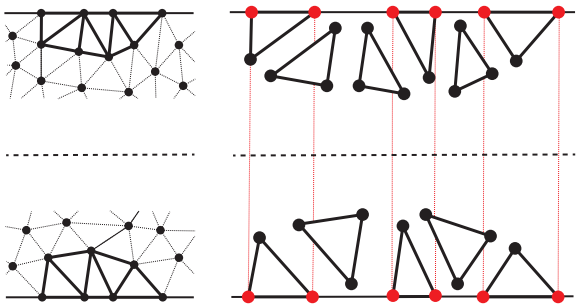


Figure 4.4: Example of mesh with the respective node pairing after cohesive elements' insertion. Only the boundary nodes (red dots) are paired and used in the PBC equations. Obtained from [92].

elements were added. When the nodes are replicated, the two edge nodes of the grey and purple faces are "downranked" to face nodes. The corner node would also be an edge node if one of the green faces belonged to another element. Although a similar methodology should have been used for corners in the previous 2D work, it was never required because the meshing software used (Gmsh) always generated a mesh where the corners ended up being split by a cohesive element, thus downranking them to the same type used for all other nodes.

With the equations for PBCs being implemented this way and requiring a pair of the same type on the opposite side, the mesh used needs to meet an unusual requirement for the equations to be properly implemented. This requirement can be summarized in the following way: prior to node replication, all free edges[1] observed along all 4 RVE edges aligned with a particular direction must have the same relative position and all nodes in the corners need to be of the same type. If the mesh doesn't meet this condition, spurious crack arrest will occur at the RVE edges, as schematically shown in Figure 4.6. Because this is a requirement that cannot be directly enforced through the meshing software, a workaround had to be implemented (see section §4.2.5) to fix a mesh in order for it to be met. In theory, it is possible to implement these equations, while avoiding spurious crack arrest, with only the free edge distribution of diagonally opposite edges matching. However, that doesn't remove the need for a mesh fixing procedure and it also makes their implementation much more cumbersome.

[1] A free edge is defined as an edge that only belongs to a single element.

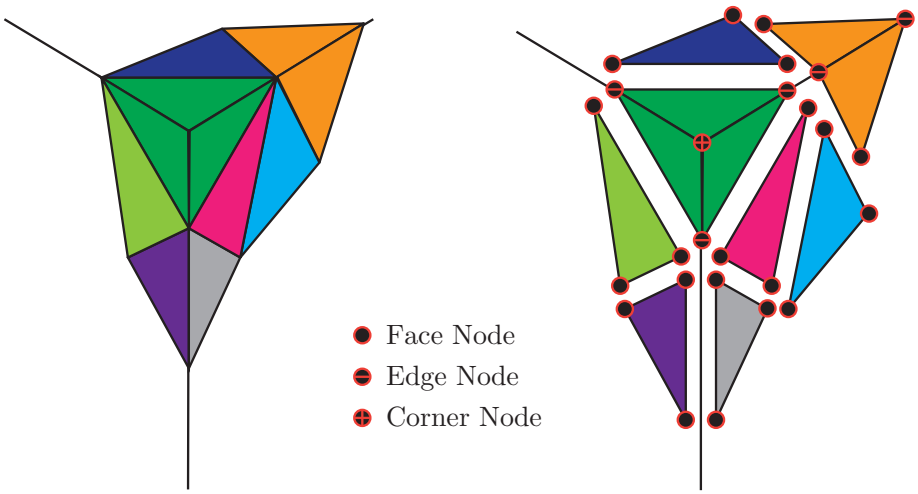


Figure 4.5: Example of a mesh (only boundary faces are shown) and the type of equations used for each node. Faces with the same color belong to the same element.

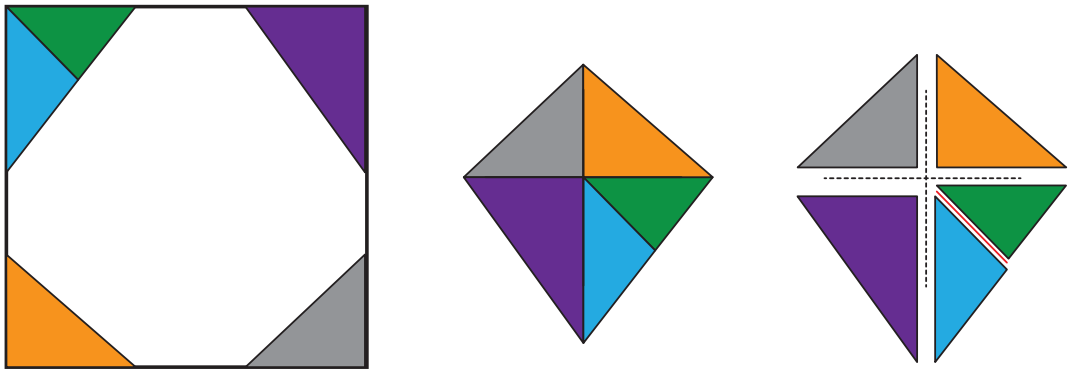


Figure 4.6: Example where a mismatch between the mesh at the edges (corners in the 2D view) leads to a spurious crack arrest. If a crack is growing past the cohesive element (red line) shown, it cannot cross the RVE boundaries as there's no adjacent cohesive element to form a continuous crack.

4.1.4. Loads

Loads are introduced in the model in the form of a displacement-based loading, according to the prescribed macroscopic strain tensor, $\overline{\boldsymbol{\varepsilon}}$. These displacements are applied on the dummy nodes, as described in equations 4.14, and the magnitudes of the tensor components are chosen such that the RVE fully breaks (i.e. forms a periodic crack).

$$\begin{aligned}
\mathbf{u}^{\mathrm{DU}}_{\mathrm{DUMMY}} &= \overline{\varepsilon}_{13}\, t\mathbf{e}_1 + \overline{\varepsilon}_{23}\, t\mathbf{e}_2 + \overline{\varepsilon}_{33}\, t\mathbf{e}_3 \\
\mathbf{u}^{\mathrm{LR}}_{\mathrm{DUMMY}} &= \overline{\varepsilon}_{11}\, w\mathbf{e}_1 + \overline{\varepsilon}_{21}\, w\mathbf{e}_2 + \overline{\varepsilon}_{31}\, w\mathbf{e}_3 \\
\mathbf{u}^{\mathrm{FB}}_{\mathrm{DUMMY}} &= \overline{\varepsilon}_{12}\, l\mathbf{e}_1 + \overline{\varepsilon}_{22}\, l\mathbf{e}_2 + \overline{\varepsilon}_{32}\, l\mathbf{e}_3
\end{aligned} \tag{4.14}$$

4.2. Software

Having so far described all details of a model at the end of pre-processing, in this section it will be explained how the software used to create these models works.

4.2.1. Overview

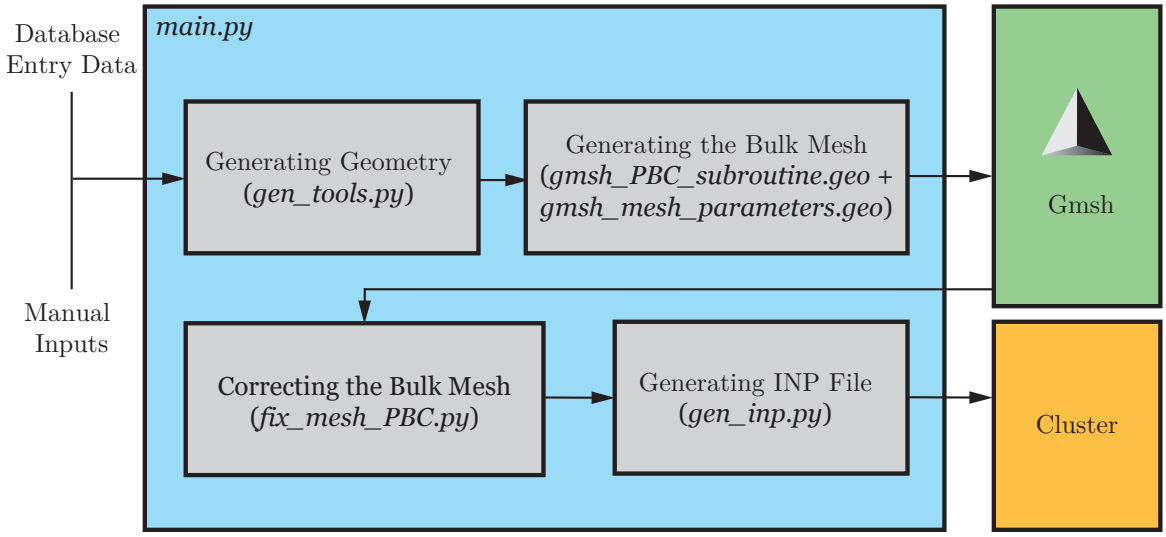


Figure 4.7: Flow diagram of pre-processing procedure.

The software architecture implemented for pre-processing can be seen in Figure 4.7. In summary, the pre-processing starts by taking the parameters stored in the respective database entry, as well as other inputs which are manually provided (such as paths or which test is to be processed) and feeding them to the main pre-processing script. This data is first used to create the geometry, with its entities being written to a .geo file at the end of it. Other routines are appended to this file, which is submitted to the meshing software Gmsh to obtain a mesh for the geometry. The resulting mesh data is then read and fixed to meet the requirements for a correct implementation of the equations for PBCs. Having a valid bulk mesh, the final step consists in inserting cohesive elements between all elements and creating the INP file that describes the model. This file is then submitted to the cluster for analysis.

4.2.2. Test Definition

For each test/analysis, the parameters that define it are stored in a dedicated entry within an SQL database. A complete list of the database parameters and their definition can be found in Appendix A.

4.2.3. Generating the Geometry

In order to perform the required analyses, a geometry consisting of a given number of fibers within a volume with the shape of a rectangular prism needs to be created (see Fig. 4.8, for example). Furthermore, with PBCs being used, every time a fiber intersects an RVE boundary, a periodic replica needs to be present on the opposite side face. Here, the algorithm implemented to create such geometry is described.

Implementation Overview

The module used to generate the geometry has an object-oriented architecture organized in 3 entity levels: RVE, Plies, Fibers. The Fibers object definition is done in another sub-module to facilitate the introduction of other types of phases (e.g. voids) in the future. In a very simplified way, the geometry generation process can be summarized as going from larger to smaller entities - first, the RVE is created, then the plies and only at the end the fibers. In order to reduce the complexity of this algorithm, only straight fibers perpendicular to the stacking direction are generated. The roles played by each entity level on the geometry generation are described in the next sections.

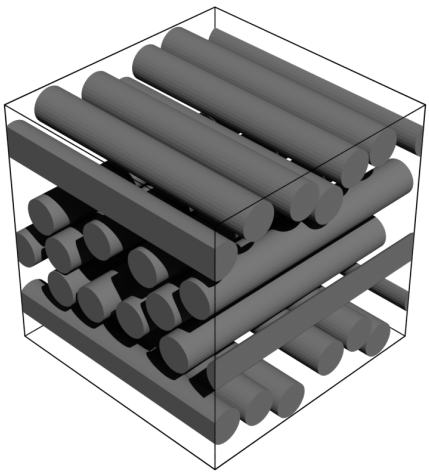


Figure 4.8: Example of a fiber distribution in an RVE with a [0/90/0] layup and $V_f = 30\%$

At the RVE level

The RVE object holds all the information regarding the geometry and is given as input for the mesh generation. Besides some entity-specific parameters (e.g dimensions, boundary conditions, ...) all other information is stored in the Ply entities that make up the RVE.

At the RVE level, geometry creation begins with determining its dimensions. For 3D RVEs that contain certain angles, size constraints arise due to the required periodicity. Using the work of Shi [88], which dealt with a similar problem when modelling slip systems in steel alloys using PBCs, and adapting it to individual fibers within a ply, it can be concluded that infinite repetitions of a straight fiber will be generated unless the RVE's dimensions are such that the distance between a fiber and its first periodic image is an integer multiple, n, as shown in Fig. 4.9. Since the goal is to obtain a fiber distribution that is random within the RVE and is only periodic at the boundary, $n = 1$ is used, giving the following governing equation for the size constraint:

$$l = \frac{w}{\tan(\theta)} \tag{4.15}$$

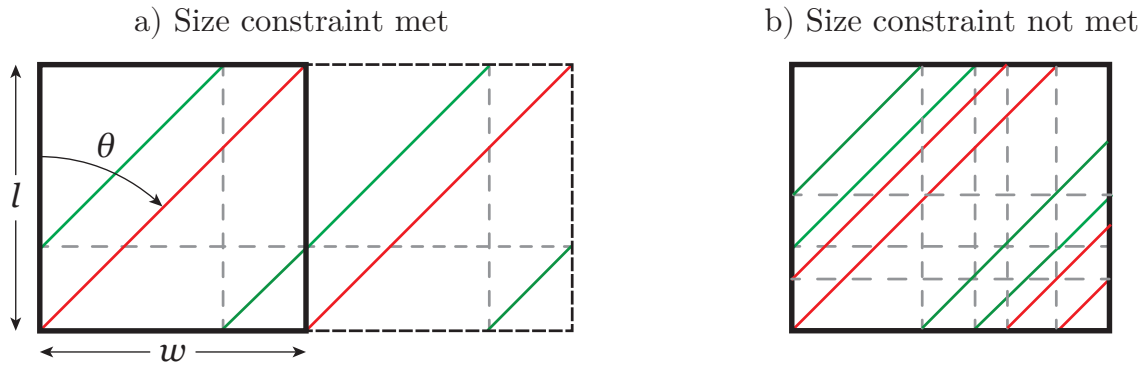


Figure 4.9: Schematic representation of how the periodic replicas of each fiber create a size constraint. In a), the bounding geometry has the correct size. For such a case, the two generic fibers (represented in green and red) cross the geometry in a way such their end point is also the starting point. For b), the size constraint is no longer met. For this case, the same fibers as before now intersect the bounding geometry in different points, such that more replicas are added (only the 2 first replicas for each fiber are represented). Since the new size used is not an integer multiple of the other dimension, regardless of how many replicas are added, no replica will ever end at the original starting point, leading to the creation of infinite replicas. Adapted from [88]

Since this relation determines the RVE's length based on the position of the next periodic image of the fiber, it's not applicable to 0° and 90° fibers, since they have no periodic images, thus allowing for an arbitrary RVE size to be used. Furthermore, although in theory the correct w can be determined for any angle θ, in practice this is not true. Due to round-off errors, the only angles besides 0° and 90° that are allowed are 30°, 45° and 60°. The symmetric values of these angles can also be used. Moreover, since different angles will enforce different RVE sizes, combinations involving different angles that are not 0° or 90° can only occur between fibers with symmetric angles (for example, if the layup already has a 45° ply, the only other angle besides 0° and 90° that is allowed is −45°). Thus, before fiber generation within each ply can proceed, the layup is evaluated to see if all the angle requirements are met and to identify which one is the driving angle regarding the RVE's size. Taking all of this into account, the 3 RVE's dimensions (see Fig. 4.3) are obtained as follows:

- **Width** (w): Prescribed directly as an input
- **Length** (l): If angles that impose a size constraint are present, it's determined from w using equation 4.15. If none are present (i.e layup only has 0° and/or 90° plies), the l_{default} value, given as input, is taken instead.
- **Thickness** (t): Determined by adding the thickness of all the plies in the layup, which in turn are specified directly as an input.

In terms of boundary conditions, there are two different sets: boundary conditions for the geometry and boundary conditions for the analysis/input file. The first ones control if fibers are allowed to cross the boundaries, while the latter define what equations are used in the Abaqus model. Most of the time, they are both periodic for all faces, but other combinations are allowed, such as wall boundary conditions for the geometry with periodic boundary conditions for the model or periodic boundary conditions for the geometry with uniform boundary conditions for the model. After determining the size of the RVE, each ply entity is created. Once completed, the geometry's boundary conditions for the whole RVE are translated into the specific boundary conditions of each ply, which depend on its relative position in the stack.

At the Ply level

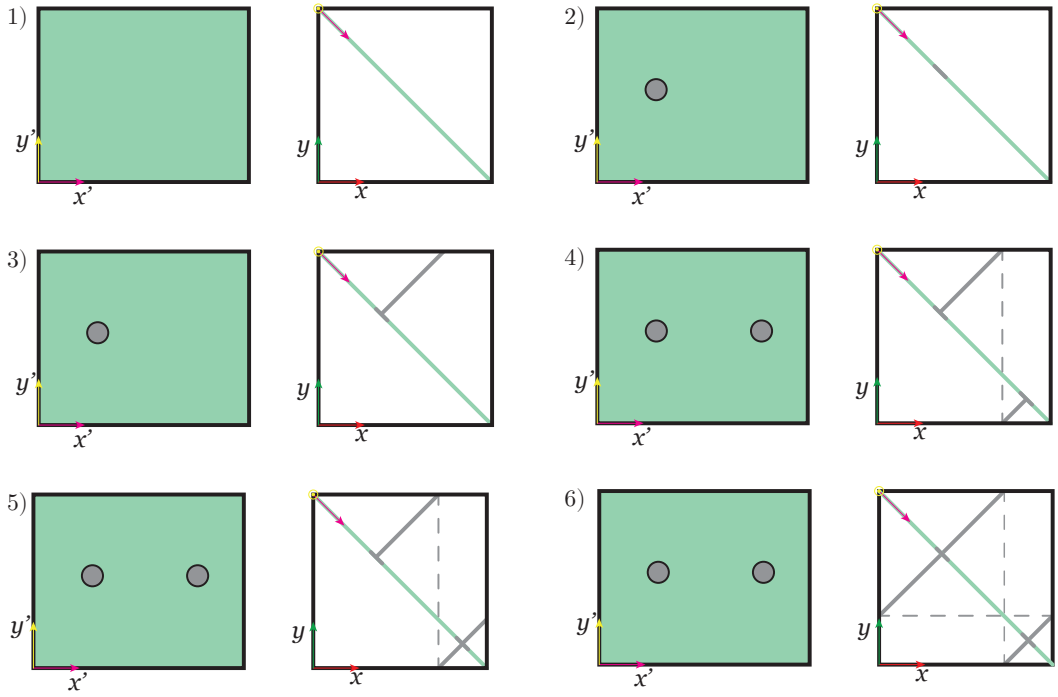


Figure 4.10: Simplified representation of the algorithm used to generate a single fiber.

Once the appropriate information has been passed from the RVE to each ply, the fiber distribution within each of them is generated. For that, a "2.5D" implementation of the Hard-Core Model([5]) has been used. It couples the sequential selection of a random point in a planar grid to be the fiber center and the extrusion of that point along the plane's normal direction to form a straight line. The way this algorithm works for a single fiber is shown in Fig. 4.10 and is as follows:

1. The generator plane (represented in green) is created. This plane is perpendicular to the fiber direction of the ply. It is formed by the base and top diagonals of the ply's box, as well as the vertical edges that connect them. In Fig. 4.10, the generator plane drawn is one used for a 45° ply. For negative angles, the other diagonal is used. For 0° the plane $y = 0$ is used and for 90° the plane $x = 0$ is used.

2. From the points that form a uniform grid over the generator plane, one of them is randomly picked. This point is then used as a fiber center.

3. From the fiber center, a straight line segment is drawn until a boundary is reached. The generator plane is also considered a boundary and the algorithm ignores the boundary where it started the extrusion as a possible intersection.

4. The periodic image of the point where the boundary was crossed is created. From there, a new segment is drawn, using this new point as a starting point, until the next boundary is found. When the boundary crossed is the generator plane, another fiber center is added to the generator plane, removing the points within the fiber cross section from the list of available points in the grid.

5. New segments are drawn from the previous point (in this case, the second crossing with the generator plane) until a new boundary is found.

6. The periodic image of the crossing point is created and used, once again, as a starting point for a new segment. When a segment stops at the initial point (first fiber center), the loop formed by the multiple segments is considered to be closed. A single fiber is considered to be a closed loop composed of one or more segments.

The number of fibers to be created for each ply is controlled by the target fiber volume fraction (V_f) of each ply, which is given as an input through the SQL database. Since all fibers have the same diameter and it is kept constant along their length, the target fiber volume fraction can be converted to a total length of segments, which is equivalent. This value is then used as a threshold during fiber generation. Furthermore, given that the fibers are straight, all fibers in a given ply have the same total (loop) length. This can be used to determine the actual number of fibers required to meet the fiber volume fraction target before generating the fibers.

As multiple fibers are added, it is necessary to make sure there's no overlap/interference between them. While fibers are being created, interference is checked for every time a fiber crosses the generator plane. This is done by evaluating if the minimum distance between the position of the current fiber center and the position of all other fiber centers already in the generator plane, as well as their periodic images, is larger than a fiber diameter.

Once the target number of fibers has been reached, the periodic images of fibers (clones) are created. These are necessary because, even though the lines that form the segments are fully contained within the RVE's volume, the fibers have a non-zero diameter, which leads to certain portions crossing the RVE boundaries if they are too close to them. Having already performed the interference detection while also checking for the potential presence of the periodic images, creating them simply consists in copying and translating segments for which the distance between the generator plane boundary and the associated fiber center is smaller than a fiber radius. Since the fiber volume fraction is determined based on total the segment length, the contribution of these periodic images to the fiber volume fraction has already been taken into account.

At the Fiber level

The Fiber level is the most basic one. It's only used to group and store information about each fiber. It also includes methods to create its geometry in Gmsh.

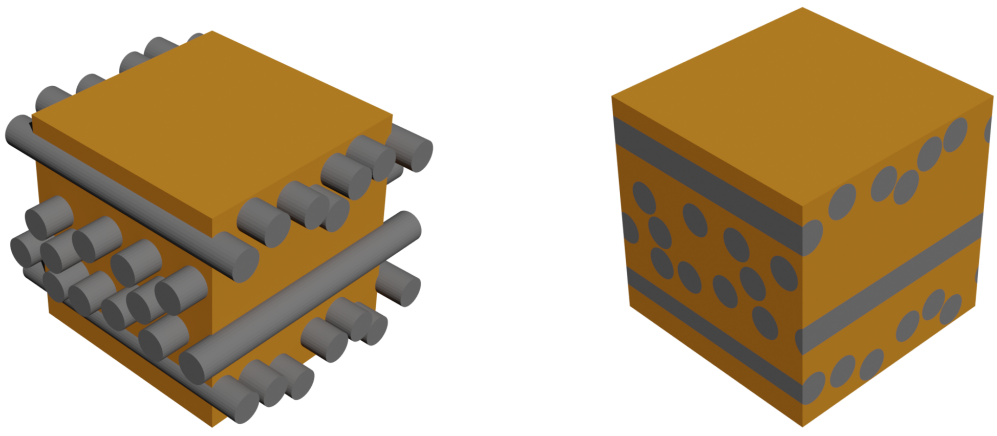
4.2.4. Generating the Bulk Mesh

Once the geometry has been created, the next step is meshing it. The open-source software Gmsh (version 4.9.3) [31] was used for this.

To generate a mesh, a .geo file needs to be provided. In it, information regarding the geometry and the meshing parameters to be used is specified. The creation of this file starts by defining the geometry. First, a box that defines the RVE volume is created. Then, for each fiber segment in the RVE, a cylinder entity is created after extending the segment to ensure that the cylinder fully crosses the RVE boundary (see Fig. 4.11). The resulting "excess" lying outside of the RVE is trimmed using the boolean operations provided by the OpenCASCADE kernel. Lastly, Physical Groups (sets of geometrical entities for which a physical group ID will be attributed to all elements associated with their geometrical entities) are created for each segment and for the matrix. Information about phase and, for fibers, also ply, fiber and segment IDs is passed through the physical group's name using a specific format.

Furthermore, if PBCs are to be used, the mesh must also be periodic. If that's the case, a dedicated routine is introduced after generating the geometry to form pairs of periodic surfaces and enforce this constraint in the meshing algorithm. The mesh periodicity is enforced using bounding boxes, which are rectangular boxes that fully contain a geometry. First, the surfaces on a given face are selected by checking if their respective bounding box is contained in the bounding box of the whole face. Then, for each of these surfaces, their bounding box is translated to the position in the opposite face where their periodic counterparts are expected to be and used to identify them. Once a match is found, the two faces are coupled using the Periodic Surface option. This new methodology is more robust than the 3D equivalent of the implementation used in 2D, since it doesn't require any *a priori* knowledge regarding the shape of the surfaces being paired. This makes it ideal for future problems where voids or fibers with cross sections that are not circular might be involved.

However, the way Gmsh creates these pairs of periodic surfaces introduces a new problem. When periodic constraints are assigned, Gmsh ensures that the mesh that will be created is periodic by copying the seeding used in each curve of a surface to its counterpart in the other surface. This becomes a problem in cases where the surfaces have the same shape and size, but the curves and points that define them don't (Fig. 4.12). For the geometries used in this thesis, such cases occur for fibers that intersect the RVE at an angle. These fibers have more than one segment, with each segment being created as a cylinder entity before trimming. While in a single cylinder, the points and curves of the two circular ends match, the same is not always true for the ends of different cylinders. Since the position of the curves and points is automatically defined by Gmsh when performing the boolean operations for trimming, there's no way to ensure that there's a point and curve match when the ends of different cylinders are paired. As a result, RVEs that contain fiber orientations other than 0° or 90° cannot be processed, since the obtained mesh will not be periodic.



(a) Matrix and untrimmed fibers

(b) Matrix and trimmed fibers

Figure 4.11: RVE geometry before and after fiber trimming.

The two surfaces have the same shape and size. The curves and points used to define them also have the same relative positions.As a result, Gmsh can enforce periodicity, since the seeding used in each curve can be passed to its equivalent in the other surface.

The two surfaces still have the same shape and size. However, the curves and points that define them no longer have the same relative positions. Gmsh is unable to enforce periodicity in this situation.

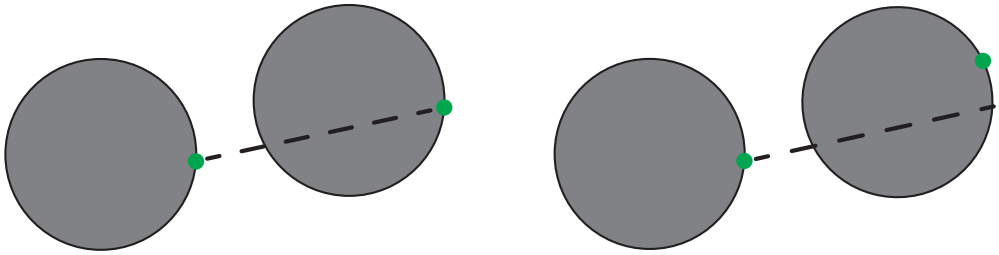


Figure 4.12: Example involving two periodic circles where the problem with enforcing periodicity through Gmsh is illustrated. Unless the curves and points that define a surface match, Gmsh is unable to assign its periodic mesh constraint to these surfaces

Lastly, the meshing parameters are defined. While the grid size is given as an input and can be changed for each test, the remaining parameters were kept constant. A 3D Delaunay triangulation algorithm is used to create a non-structured mesh of linear tetrahedrons. Similarly to what was done for 2D, 10 mesh smoothing steps were used and the minimum and maximum element sizes are set to half the grid size and the grid size, respectively.

4.2.5. Correcting the Bulk Mesh

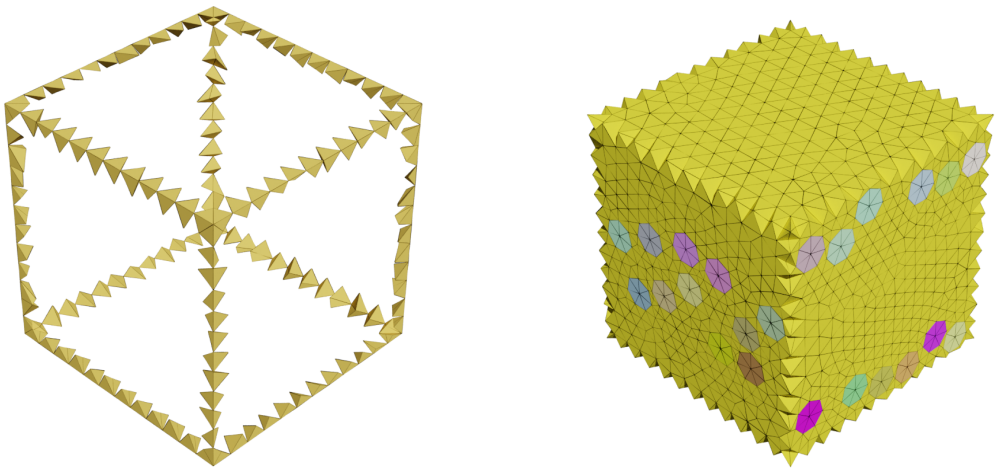
As mentioned before, in order to apply the equations for PBCs, the mesh used needs to meet some requirements. Since one of them cannot be directly enforced through Gmsh, the mesh needs to be corrected separately. At the end of this process, the mesh can be seen as being two separate meshes combined: the scripted mesh (Fig. 4.13a) and the shell mesh (Fig. 4.13b). Both of them use the element faces that define the RVE boundary in the mesh created in section 4.2.4 as a starting point. The scripted mesh is obtained by creating a mesh at the edges that meets the periodicity requirements using pairs of faces that share 2 nodes at an edge. On the other hand, the shell mesh is obtained by using those edges that define the RVE boundary and replacing the faces used to create the scripted mesh with the internal faces of its elements. The result is a hollow polyhedron with triangular faces - the shell - which fills the remaining volume and is given to Gmsh for automatic meshing.

The mesh fixing process starts by reading the mesh data and identifying the boundary faces on each of the RVE's faces. This information is then used to identify the edge and corner nodes. From this, segments are created, which are lines formed by 2 consecutive edge nodes of the same RVE edge. The segments with the same relative position in edges aligned with a given direction are grouped to form segment groups. Lastly, the situation at each segment is assessed. For each segment there are two possible cases:

- **Case 1:** The faces that form that segment belong to the same element.
- **Case 2:** The faces that form that segment belong to the different elements.

The end goal of this mesh fixing procedure is to correct the initial mesh in a way that all segment groups have the same case. The last step performed before proceeding with the scripted mesh generation is identifying the interface faces. These are faces shared by elements belonging to different physical volumes. This is achieved by first identifying the common faces between all elements (3 nodes shared by 2 elements). Interface faces are then filtered by selecting common faces where the associated elements belong to a different physical volume.

The scripted mesh generation begins by assigning an equal target case for each segment in a group. When all the segments in a group already have the same case, the target case is the one already observed or case 2 if one of the segment's nodes is a corner. When the segments in a group have different cases, case 2 is forced. In both situations, case 2 is enforced because a transition from case 1 to 2 is easier to handle than a transition from case 2 to 1, as it will be shown later. After this, the situation in each segment is handled based on the observed transition - the comparison between the old case and the new (target) case. Each transition is handled as follows:



(a) Scripted mesh. (b) Shell mesh

Figure 4.13: Scripted and shell meshes, which form the fixed mesh when combined.

- **Case 1 to case 1 transition:** The case is preserved. Elements are copied to the new mesh.
- **Case 1 to case 2 transition:** The element has to be broken in 2. This is done by creating a node at the midpoint of the edge that connects the node that doesn't belong to the segment in each boundary face. Each element is then created from the set of 4 nodes formed by this new node and the 3 that define its associated boundary face.
- **Case 2 to case 1 transition:** Currently not allowed. Although it could be handled by creating a single element from the nodes of the two boundary faces (6 nodes in total, but only 4 are unique), there are a few problems. One of them is that the elements whose boundary faces define the segment being handled don't necessarily share a face, which significantly complicates all the subsequent steps of the mesh fixing procedure. This transition also requires the deletion of a node, which might be used in describing an interface, thus requiring extra checks while performing the transitions.
- **Case 2 to case 2 transition:** The case is preserved. Elements are copied to the new mesh.

In the end, the scripted mesh consists in all the elements which were either copied or created while performing the transitions.

After generating the scripted mesh, the shell mesh follows. Here, a shell is considered to be a hollow polyhedron with triangular faces. The shell mesh part consists of different shells (one for each physical volume) combined together to form the remaining RVE mesh. Its creation starts by making a global shell using all the boundary triangles, which combined form the RVE's outer surface. From these, the faces used to create the scripted mesh are removed and replaced with the internal faces of the scripted mesh (i.e faces that are not boundary and that are not shared by elements of the scripted mesh). The resulting shell completely fills the RVE volume when combined with the scripted mesh. While removing and replacing faces, special care must be taken to keep track of where interfaces between different physical volumes are. The implemented algorithm does this by taking advantage of the fact that the new elements are self-contained in the old elements to match faces from one mesh to the other. Once the faces have been removed and replaced, the larger shell is broken into individual shells for each physical volume. This is achieved by filtering the faces based on the physical volume tags of their adjacent elements and on whether they are boundary or interface faces. However, not all of the faces identified during this process form a shell. In some cases, such as the one shown in Figure 4.14, faces that are not required for the shell volume definition are also present after the filtering is done. Such faces can be easily identified and removed, since they have at least one free edge (an edge shared

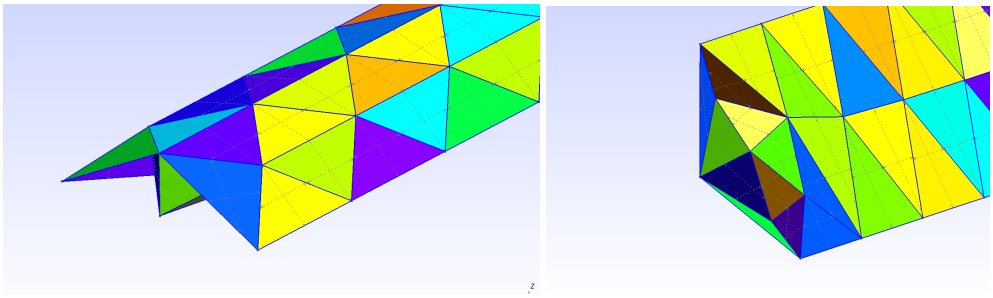


Figure 4.14: Shell where 2 free faces can be seen (colored in blue and green). It is not possible to generate 3D elements from these faces and as a result, they need to be identified and removed.

only by a single face), which can never occur in a closed shell. The faces(triangles) and points that describe them are then written to a .geo file. In it, each triangle is first created using independent IDs for the entities that define it (for example, a point shared by 2 different triangles will be created twice with a different ID). The shell is then closed by using the "Coherence" option, which automatically combined entities with redundant definitions. After this, each shell is meshed separately using Gmsh once again. Once finished, the mesh of each shell is read and given the respective physical volume tag.

The final mesh is obtained by sequentially combining the scripted mesh with the individual shell meshes. As this is done, the nodes and elements on each mesh are renumbered to ensure that their numbering ranges don't overlap. In the end, duplicate nodes can be found at the interfaces between the different meshes. These are merged and all nodes are once again renumbered, as subsequent scripts assume a continuous numbering of nodes and elements. Lastly, the order in which nodes are specified in the elements' definition is corrected, as Abaqus follows a specific convention: all face normals need to point inwards. If not respected, faces will not be displayed correctly.

In order to avoid creating an overly complex algorithm, some simplifications were introduced which resulted in the algorithm having some limitations. All of them come as a consequence of the algorithm assuming that the way the mesh is fixed on a given edge doesn't influence the way that is done for another edge. This breaks down at the corners, where interference will occur unless the following conditions are met:

- A boundary face can only be associated with a single segment.
- At a corner, 3 segments meet. The elements associated with the boundary faces of these segments cannot be in contact with each other, even if they are associated with the same segment.

Currently, the only way to obtain a mesh that fulfills these requirements is by trying repeatedly to create a new geometry and a new mesh until the mesh before fixing meets these conditions. The unintended consequence of this is that a mesh refinement cannot be performed without changing the geometry, since there's no guarantee that if the mesh meets these for a given mesh size, that it will also meet them for a different one.

4.2.6. Generating the INP file

The model is described for analysis in the form of an INP file. Before writing its data to such a file, cohesive elements are inserted between all bulk elements. To do this, common faces are first identified using the same method applied during the mesh fixing process. Then, the elements are separated from each other. For this, new nodes are created and replaced in the elements' definition in a way such that every element has a different node ID for a node that was previously shared by multiple elements. Lastly, cohesive elements are created using the 6 nodes (3 for each element's face) that can now be found at each common face. While the node numbering had to follow an anti-clockwise convention in 2D to specify the thickness direction correctly, this is not required in 3D. Instead, for each shared face and after the nodes have been replicated, the cohesive element is defined by taking the nodes of one face, which will become the nodes 1 to 3 of the cohesive element (Fig. 4.2), and then taking their respective duplicates in the same order, which will become the element's nodes 4 to 6.

Periodic boundary conditions are also applied. For that, boundary faces are, once again, identified. From these, pairs of periodic nodes on opposite faces are formed. Then, the nodes are organized based on which type of equations apply to them (face, edge or corner).

At last, with the model data organized, it is written to the INP file respecting the formats specified in the documentation for each parameter.

5

Numerical Implementation: Processing and Post-processing

5.1. Post-processing procedure overview

As explained in chapter §3, having solved the boundary value problem, the data needs to be separated in order to identify the periodic crack(s) and its/their response isolated from that of the bulk material to be homogenized separately.

However, it was not possible to fully implement such a workflow in its entirety. The first reason for this was the delay caused by the need to implement the mesh fixing step. The other was the lack of good data to properly debug the scripts where the algorithms for 3D were implemented, due to reasons that will be explained in more detail in the results chapter. Consequently, an alternative post-processing scheme was used. In the absence of data where the crack has been identified and its response isolated, the elastic response of the bulk material was used.

In the next section, the simplified approach used to post-process the results presented in chapter §6 is first introduced. Then, the newly developed but still untested algorithms for the conventional post-processing procedure are described. Some suggestions for possible solutions to the parts which have yet to be implemented are also given.

5.2. Simplified post-processing

As mentioned before, the conventional homogenization procedure requires the identification of which cohesive elements constitute the periodic crack. Since the dedicated algorithm to perform such a task has yet to be implemented and tested in 3D, an alternative possibility to analyse the material's fracture behaviour and assess how certain mechanisms influence it is looking at the elastic response of the bulk elements in the model. Although it's not ideal, it's nonetheless possible to get a general idea of what an ETSL obtained through the conventional post-processing would look like.

The first step in homogenizing the bulk elements' data is extracting it from the .odb file obtained at the end of an analysis. For every time step, the required elemental quantities are obtained through equations 5.1, 5.2 and 5.3. While the volume of an element, v_{elem}, is a scalar and is provided directly, the stress, $\boldsymbol{\sigma}_{\text{elem}}$, and strain tensors, $\boldsymbol{\varepsilon}_{\text{elem}}$, are extracted as a vector and need to be reshaped. Furthermore, as noticed by Suárez-Millán [92], the shear strain components extracted from Abaqus are actually engineering strains, which need

to be divided by 2.

$$\boldsymbol{\sigma}_{\text{elem}} = \begin{bmatrix} \texttt{S11} & \texttt{S12} & \texttt{S13} \\ \texttt{S12} & \texttt{S22} & \texttt{S23} \\ \texttt{S13} & \texttt{S23} & \texttt{S33} \end{bmatrix} \tag{5.1}$$

$$\boldsymbol{\varepsilon}_{\text{elem}} = \frac{1}{2} \begin{bmatrix} \texttt{2LE11} & \texttt{LE12} & \texttt{LE13} \\ \texttt{LE12} & \texttt{2LE22} & \texttt{LE23} \\ \texttt{LE13} & \texttt{LE23} & \texttt{2LE33} \end{bmatrix} \tag{5.2}$$

$$v_{\text{elem}} = \texttt{EVOL}_{\text{elem}} \tag{5.3}$$

$$\tag{5.4}$$

From these quantities, the volume averaged stress and strain tensors for the bulk domain are then computed at each time step using equations 5.5 and 5.6. These are the numerical versions of equations 3.15 and 3.12, respectively.

$$\langle \boldsymbol{\sigma} \rangle_{\Omega} = \frac{1}{|\Omega|} \sum_{\text{elem} \in \Omega} \boldsymbol{\sigma}_{\text{elem}} v_{\text{elem}} \tag{5.5}$$

$$\langle \boldsymbol{\varepsilon} \rangle_{\Omega} = \frac{1}{|\Omega|} \sum_{\text{elem} \in \Omega} \boldsymbol{\varepsilon}_{\text{elem}} v_{\text{elem}} \tag{5.6}$$

where $|\Omega|$ is the total domain volume. It is obtained by adding the volume of all bulk elements at that given time step:

$$|\Omega| = \sum_{\text{elem} \in \Omega} v_{\text{elem}} \tag{5.7}$$

The averaged quantities are then plotted as a function of the applied strain at the respective time instant, which is given by:

$$\overline{\boldsymbol{\varepsilon}}(t) = t\overline{\boldsymbol{\varepsilon}}_{t=1} \quad \text{for} \quad t \in [0,1] \tag{5.8}$$

5.3. Full post-processing - Proposal of possible implementation

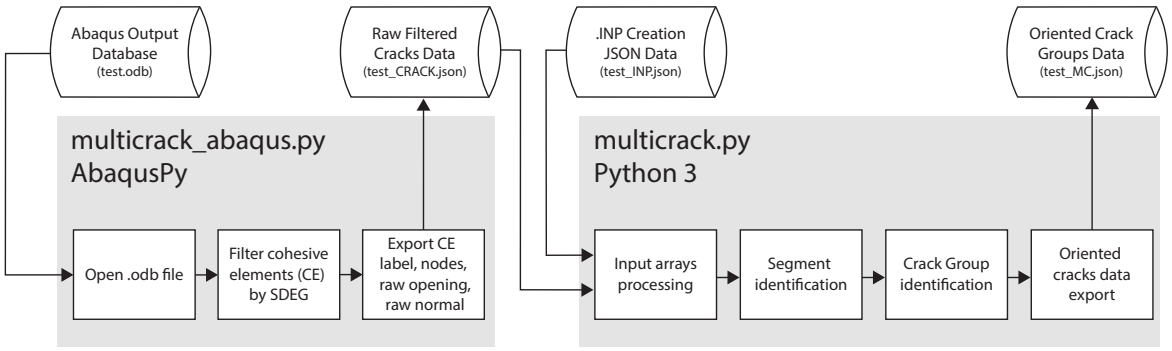


Figure 5.1: Code structure for the conventional post-processing's first step. Obtained from [92].

The complete software structure of the proposed conventional post-processing is the same one used in the 2D implementation [92], but the individual scripts are modified to include the necessary changes. The conventional procedure is divided in two major steps. The first one, shown in Figure 5.1, consists in processing the data associated with the cracked part of the domain. It starts with the preparation of the data associated with the cohesive elements, which is then used to identify where the periodic cracks are. In the second step, the oriented cracks' data is used in combination with the data for the bulk part of the domain and the relevant volume and surface integrals are computed.

5.3.1. Crack normal reorientation and data preparation

In the first step of post-processing, the crack is identified. The first part of this step consists in reading the cohesive element's data from the .odb file using a dedicated module and preparing it to be used by the crack identification module in the second part of this step.

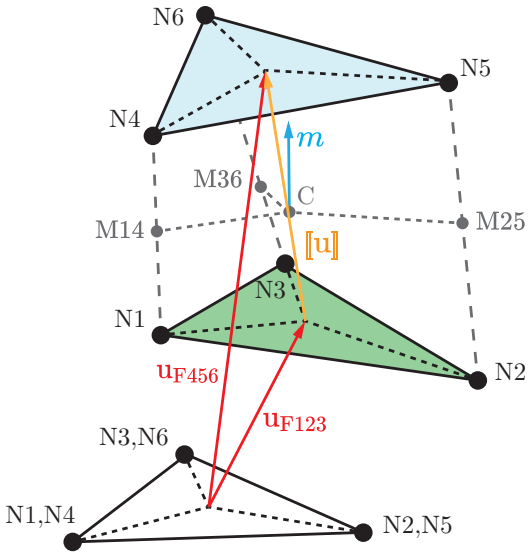


Figure 5.2: Opening displacement and normal vectors as a function of the nodal displacements.

This module works in the following way. For each time interval, it iterates through all cohesive elements in the model. Data is extracted from cohesive elements for which the damage variable (`SDEG`) is larger than the specified cut-off value, with the remaining being ignored. For each element considered, on the last time frame, the following quantities are stored: element and node IDs, frame time, damage variable, section, opening displacement vector and normal vector. All of these, except the last two, can be extracted directly, while the others are computed from the nodal displacements shown in Fig. 5.2.

The opening displacement, $[\![\mathbf{u}]\!]$, is obtained as follows:

$$\mathbf{u}_{\mathrm{F123}} = \frac{1}{3}\left(\mathbf{u}_{\mathrm{N1}} + \mathbf{u}_{\mathrm{N2}} + \mathbf{u}_{\mathrm{N3}}\right) \tag{5.9}$$

$$\mathbf{u}_{\mathrm{F456}} = \frac{1}{3}\left(\mathbf{u}_{\mathrm{N4}} + \mathbf{u}_{\mathrm{N5}} + \mathbf{u}_{\mathrm{N6}}\right) \tag{5.10}$$

$$[\![\mathbf{u}]\!] = \left(\mathbf{u}_{\mathrm{F456}} - \mathbf{u}_{\mathrm{F123}}\right) * f_{\mathrm{u}} \tag{5.11}$$

where f_{u} is the flip function, defined as:

$$f_{\mathrm{u}} = \begin{cases} 1 & \text{if} \quad |\mathbf{u}_{\mathrm{F456}}| > |\mathbf{u}_{\mathrm{F123}}| \\ \mathrm{ind} & \text{if} \quad |\mathbf{u}_{\mathrm{F456}}| = |\mathbf{u}_{\mathrm{F123}}| \\ -1 & \text{if} \quad |\mathbf{u}_{\mathrm{F456}}| < |\mathbf{u}_{\mathrm{F123}}| \end{cases} \tag{5.12}$$

The flip function ensures that the opening displacement vector always points from the side with the smallest displacement to the side with the largest. When both sides have the same displacement, the flip function is indeterminate. When this happens, the zero vector is taken as the opening displacement vector.

As for the element normal, it is obtained in the following way:

$$\mathbf{u}_{\mathrm{M14}} = \frac{1}{2}\left(\mathbf{u}_{\mathrm{N1}} + \mathbf{u}_{\mathrm{N4}}\right) \tag{5.13}$$

$$\mathbf{u}_{\mathrm{M25}} = \frac{1}{2}\left(\mathbf{u}_{\mathrm{N2}} + \mathbf{u}_{\mathrm{N5}}\right) \tag{5.14}$$

$$\mathbf{u}_{\mathrm{M36}} = \frac{1}{2}\left(\mathbf{u}_{\mathrm{N3}} + \mathbf{u}_{\mathrm{N6}}\right) \tag{5.15}$$

$$\mathbf{u}_{\mathrm{C}} = \frac{1}{3}\left(\mathbf{u}_{\mathrm{M14}} + \mathbf{u}_{\mathrm{M25}} + \mathbf{u}_{\mathrm{M36}}\right) \tag{5.16}$$

$$\boldsymbol{m}^{*} = \left(\left(\mathbf{u}_{\mathrm{M14}} - \mathbf{u}_{\mathrm{C}}\right) \times \left(\mathbf{u}_{\mathrm{M25}} - \mathbf{u}_{\mathrm{C}}\right)\right) \tag{5.17}$$

$$\boldsymbol{m} = \boldsymbol{m}^{*} * f_{\mathrm{m}} \tag{5.18}$$

where f_{m} is the flip function for the normal, defined as:

$$f_{\mathrm{m}} = \begin{cases} 1 & \text{if} \quad \boldsymbol{m}^* \cdot [\![\mathbf{u}]\!] > 0 \\ f_u & \text{if} \quad \boldsymbol{m}^* \cdot [\![\mathbf{u}]\!] = 0 \\ -1 & \text{if} \quad \boldsymbol{m}^* \cdot [\![\mathbf{u}]\!] > 0 \end{cases} \tag{5.19}$$

This flip function orients the normal in such a way that $\boldsymbol{m} \cdot [\![\mathbf{u}]\!] \geq 0$ is always verified. When the flip function f_{u} is indeterminate, the normal vector is also taken as the zero vector.

In the end, the cohesive element's filtered data is exported to be used in the crack identification module.

5.3.2. Crack Identification

Having filtered the cohesive element's data and computed the opening and normal vectors of each element, the cracks within this data need to be identified, since only its periodic part is relevant to the homogenization procedure [98]. This constitutes the second part of the first post-processing step and is handled by a dedicated Python module, as shown in Fig. 5.1.

In the 3D implementation, this module follows the same steps as its 2D counterpart [92], but with significant changes. It starts by combining the filtered cohesive elements into segments. A segment is a set of elements that form a surface, which is either delimited by intersections with other segments or free ends. Having identified the segments, these are grouped into crack groups, which can either be sets of segments that form a main periodic crack or simply a disconnected group of isolated segments and branches.

Prior to both of these steps, node IDs are reconverted to their original IDs and the data is updated to treat cohesive elements as triangles defined by the nodes in a single face on the reference configuration.

Segment Identification

After this, segment identification is performed. In order to understand it, a key concept - the boundary status - must be introduced first. A segment's boundary is composed of a set of edges, which are referred to as boundary edges. Each boundary edge has a boundary status, which can take one of the following values (Fig. 5.3):

- **Free/Closed:** the boundary edge is only shared by a single cohesive element, which is already part of the segment.
- **Intersection:** the boundary edge is shared by more than two elements.
- **Open:** the actual status of the boundary edge has yet to be determined i.e, it can still become an intersection, a free/closed edge or just be shared by two elements, in which case the element is added to the segment.

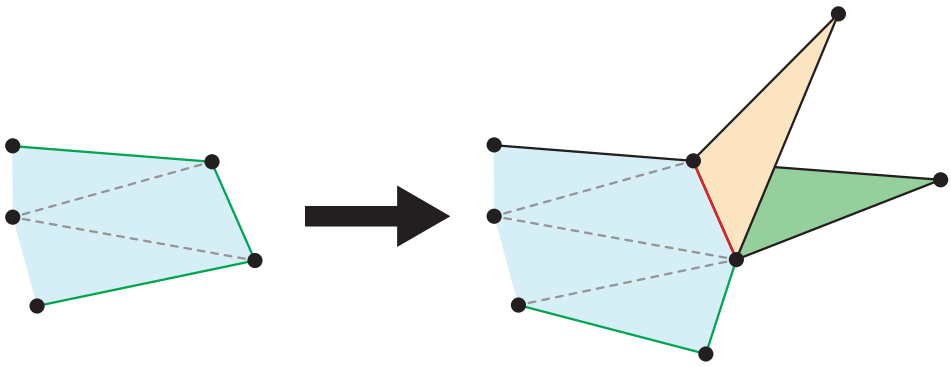


Figure 5.3: Illustration of the different boundary statuses and how they can change while the segment is growing. All triangles represent a cohesive element in the reference configuration. Cohesive elements with the same color belong to the same segment. Boundary edges with an "Open" status are marked as green, "Free/Closed" as black and "Intersections" as red. Internal edges are marked as grey dashed lines. In this example, the 3 boundary edges illustrated all start as "open". After one loop, the first one has been identified to be free/closed, the second as an intersection. The third one is only shared by two elements, and as a result, the new element is added to the segment. This "open" edge is turned into an internal edge and the new edges are marked as "open" for the next loop to evaluate.

The general idea of the segment identification algorithm consists in picking an element which initializes the segment and iteratively adding adjacent elements to it, based on the status of its boundary edges. A segment is considered to be finished when all of its boundary edges are either free/closed or intersections. New segments are created until all elements belong to a segment. The segment identification algorithm is as follows:

- Iterate over all filtered cohesive elements:
    - If the element is part of a segment already, ignore it.
    - If the element isn't part of a segment yet, a segment is initialized with that element
        - Mark the element as part of the new segment
        - Identify the IDs of the element's edges and make them the boundary edges with an "open" boundary status
        - Make a backup of the current status of each boundary edge.
        - While there are open edges (in the edges of the backup):
            - Iterate over all boundary edges that are open in the backup:
                - See how many filtered cohesive elements share that edge, besides the one associated with the edge being evaluated
                - If no other element shares this boundary edge:
                    - Update the boundary edge's status to "free/closed".
                - If one element shares this boundary edge:
                    - Add this other element to the segment.
                    - Identify the IDs of the new element's edges.
                    - For each of those edges, check if they are boundary edges already:
                        - If they aren't boundary edges:
                            - Make them and set their status to "open"
                        - If they are:
                            - Check their status:
                                - If it's free/closed, raise error, since this shouldn't be possible
                                - If it's an intersection, do nothing, since the segment is already correct in terms of elements and boundary
                                - If it's open, remove it from the boundary since its now an internal edge
                - If more than one shares this boundary edge:
                    - Update its status to intersection.
                    - Add all elements that share the edge (including the current one) to the incidence matrix for intersections
            - Update the backup of which edges are boundary edges and their status
            - Check if there are still open edges
        - Create a segment object with data describing the segment (ID, elements, boundary edges, intersections)
        - Add the segment object to the list of segments

Crack Group Identification

Once all elements have been assigned to a segment, these are grouped into crack groups, some of which will form main periodic cracks, which is the relevant part of the crack for homogenization. The 2D implementation by Suárez-Millán [92] adds segments to crack groups based on orientation, taking advantage of the fact that the maximum number of main cracks that can exist in 2D is two. This happens because main cracks only form in the longitudinal and transverse fiber directions and because the only two fiber directions that can be analyzed in 2D are 0° and 90°. Although this algorithm is quite reliable in 2D, it doesn't work as well in

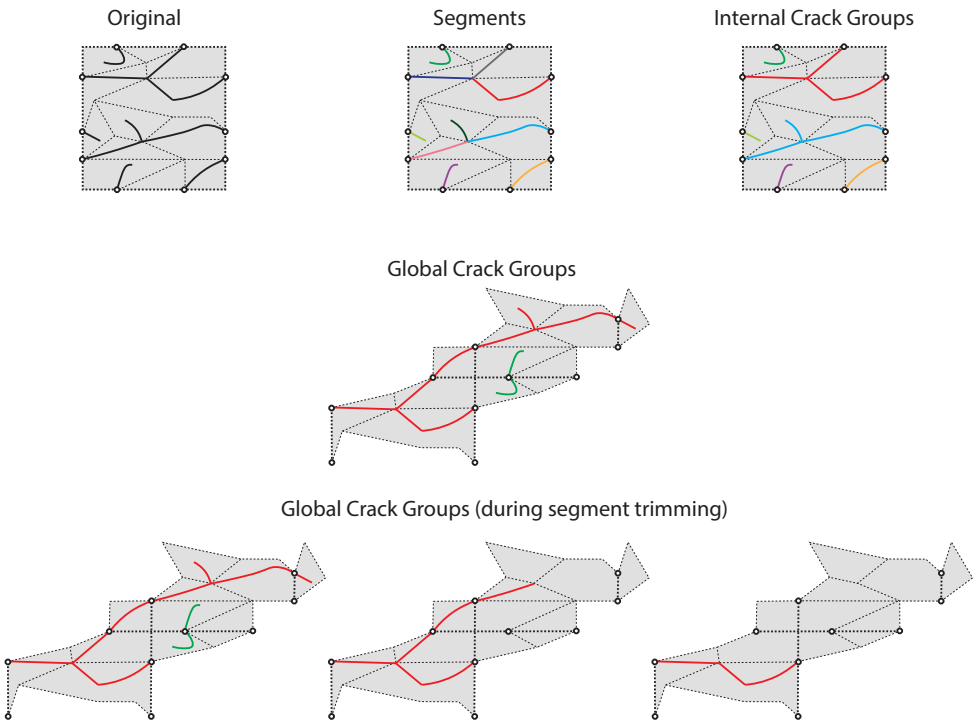


Figure 5.4: Extraction of a crack's periodic part from a fully-cracked domain. For each image, different colors are used to identify different entities according to the respective labels. The crack pattern was obtained from [98].

3D, since more fiber orientations are now possible. Instead, the proposed algorithm for 3D performs the full shifting mapping, as originally proposed in Turteltaub et al.[98]. Once this is done, the periodic cracks are extracted by identifying periodic boundary loops and iteratively growing the main crack inwards after removing all branches.

The algorithm's first step is the formation of internal crack groups (ICGs). These are groups of adjacent segments delimited only by the free/closed edges and RVE boundary edges (an edge that lies in one of the RVE faces) of its segments. These are obtained in a relatively simple way: after initializing it with a segment, new adjacent segments are added until there are no more adjacent segments left. New ICGs are created until all segments belong to an ICG. Segments that are only connected after the shifting is carried out are not considered adjacent to each other during an ICG's creation. An example of how ICGs are obtained from segments can be found in Fig. 5.4.

After this, the individual ICGs are shifted to form global crack groups (GCGs), as shown in Fig. 5.4 for example. First, the information regarding the node periodicity is used to create its equivalent for edge periodicity. A GCG is initialized by picking an ICG which doesn't have one yet. This first ICG is used as a reference for the remaining shifting associated with that GCG. Its boundary edges which are also RVE boundary edges are picked and their periodic images are identified. For each ICG associated with a periodic image which is not yet part of the current GCG, the shifting is performed on the whole ICG by by marking the ICGs, segments and edges as adjacent in the relevant arrays (no node merging or renumbering is done) and computing an offset to be later applied on the nodes' coordinates. The algorithm was only implemented up to this point. All further discussion is simply a general description of what was planned.

The step that follows is loop identification. In 3D, the concept of periodic crack extends from being a curved path with periodic end points to a curved surface with a periodic loop as boundary. This loop is formed by RVE boundary edges that are also part of the GCG's boundary. The loop being periodic means that the periodic image of an edge in the loop is also part of that same loop. From each loop, a main crack group will be created, although it is not always guaranteed that it will eventually become a periodic main crack. However, the existence of at least one periodic loop is a requirement for a volume element to be considered as fully-failed. As explained before, the number of periodic main cracks is not known *a priori*, the individual loops must be extracted from the set of edges that meet the previously mentioned requirements without

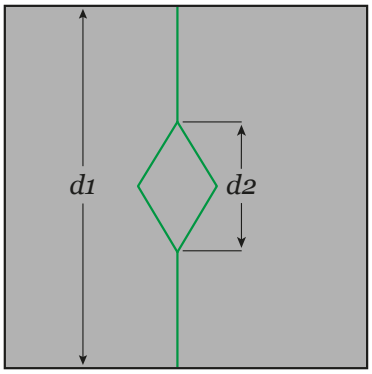


Figure 5.5: Periodic main crack with a small region where it separates in two segments that later reconnect. Since the length of this separated region, $d2$, is small enough when compared with the crack length, $d1$, the interpretation that this is a single crack that briefly separates in two is reasonable. However, if $d2$ becomes larger, eventually it will become two separate cracks that briefly overlap (imagine, for example, that the overlapping and separated regions shown were swapped). The point where the transition occurs is hard to establish and has a direct influence on the number of loops that will be identified.

the knowledge of how many loops are actually present. This is an especially hard task due to the degree of ambiguity associated with distinguishing between two separate cracks that briefly overlap and a crack that briefly separates in two (see Fig. 5.5).

The proposed algorithm for loop identification begins with grouping the individual edges into loop segments that are easier to work with. For this, the 2D segment creation algorithm can be used. Once this is done, loop segments that are branches are trimmed, since the loop must be closed. After that, an intersection point is picked and all possible paths are followed simultaneously. Every time a new intersection is found, a new alternative path is created. During this process, some paths will become invalid, such as going over the same loop segment but in the opposite direction or not actually being periodic. A loop can only be considered complete once it returns to the point where it started. In the end, only the boundary loops remain.

Once the loops have been identified, segments are trimmed from the global crack groups if they are branches (see Fig 5.4). As more segments are trimmed, other segments will become branches, until only the segments that will form the main periodic cracks remain. The main crack groups are built by picking the crack segments associated with the periodic loop and growing the crack inwards. Care must be taken while doing such inward expansion because there are segments which are shared by multiple cracks. The way such segments are handled has yet to be established, but one possibility is a weighted, shared contribution to the relevant quantities between all main periodic cracks that share them, possibly based on the angle.

After this, the remaining procedure is identical to that used in 2D. With the main periodic cracks identified, the cohesive element normals can be reoriented in a consistent way. Then, the area, traction and opening of each element can be determined. From these, the effective quantities that define the effective macroscopic crack are extracted. The orientation of the macroscopic crack can be obtained by averaging all oriented normals of the cohesive elements that form a main periodic crack. On the other hand, the area can be determined by projecting the area of its main periodic crack (obtained after shifting mapping) to a plane with normal equal to the crack-averaged normal. The effective traction is obtained by integrating the traction over the crack and the effective opening is obtained through the energetically consistent relation. Lastly, in order to verify that the Hill-Mandel condition for the crack is met, the power dissipated at both scales is obtained using the respective relations.

Lastly, note that the way artificial compliance was dealt with by Suárez-Millán [92] also assumes the presence of only two main periodic cracks, and thus needs to be revised.

6

Results

In this chapter, the results obtained for the most relevant layup and load case combinations tested are presented. Unfortunately, only limited conclusions could be extracted from the results, due to a strange numerical artifact that will be described in more detail later. For this reason, the end of the chapter is reserved for some remarks about its possible causes, as well as ways each of them can be tested and fixed in the future.

6.1. Individual Test Results and Discussion

With the pre-processing setup in place, some analyses were performed. As mentioned before, it was not possible to test orientations different than 0° and 90° due to limitations of the meshing software. Nonetheless, the two available directions were tested both individually, in unidirectional layups, and together, in multidirectional layups. In total, 14 analyses were performed in TU Delft's HPC cluster. This is a relatively small number of analyses compared with what's usually needed for verification and to draw relevant conclusions about the material's behaviour. The main reason for this was the limited number of Abaqus tokens combined with the large computational cost associated with each analysis. Since each analysis required 1 node and 4 CPUs, only 3 could be performed simultaneously without exceeding the budget for Abaqus tokens allocated to a single user of the cluster. As for the computational cost, even with a very coarse mesh and a relatively small node count (usually between 50000 and 75000 after adding the cohesive elements), each analysis could take up to 2 weeks to finish.

6.1.1. Unidirectional layups

Unidirectional layups are made up of a single ply. The two possible directions, 0° and 90°, were tested, even though they are equivalent. For this type of layups, multiple load cases were considered - transverse, longitudinal and biaxial stretching. Compression and shear were not tested because they require a contact algorithm, which has yet to be implemented. The individual tests performed for each load case will be discussed next.

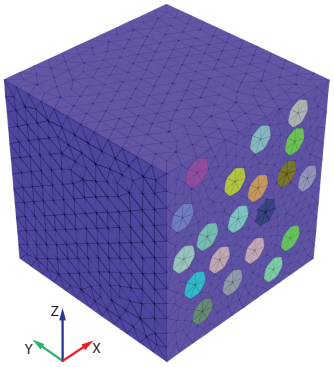


Figure 6.1: Geometry and mesh used in tests #8, #9, #10, #12 and #13, with a [0] layup.

Transverse Loading

Using the geometry and mesh shown in Figure 6.1, tests #8 and #9 were performed, where uniaxial loadings along the X and Z directions were respectively applied. Only the results from test #8 are shown, since the results from test #9 are identical. A table with the relevant parameters used for these and all other tests can be found in Appendix B.

The contour plots for the stress component aligned with the applied load and the damage variable in test

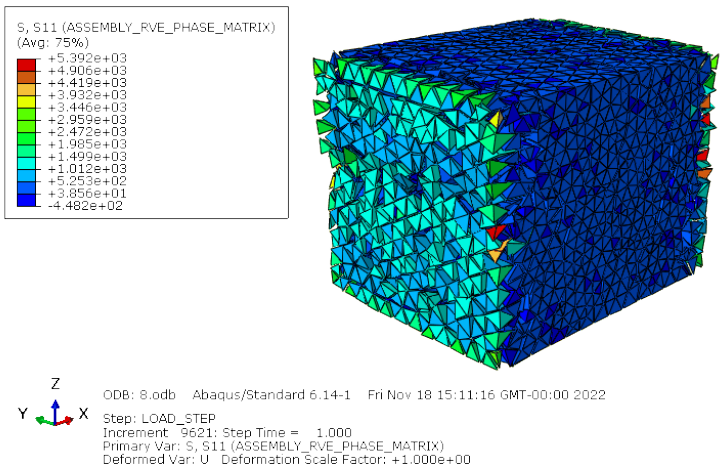


Figure 6.2: Test #8 results: S11 component of the stress tensor on all bulk elements.

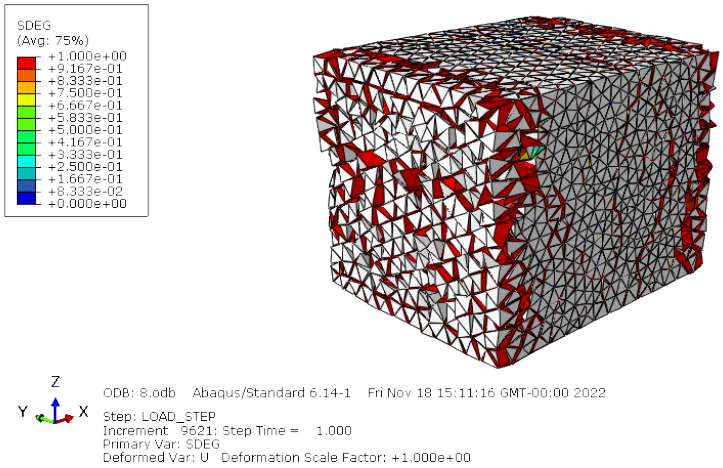
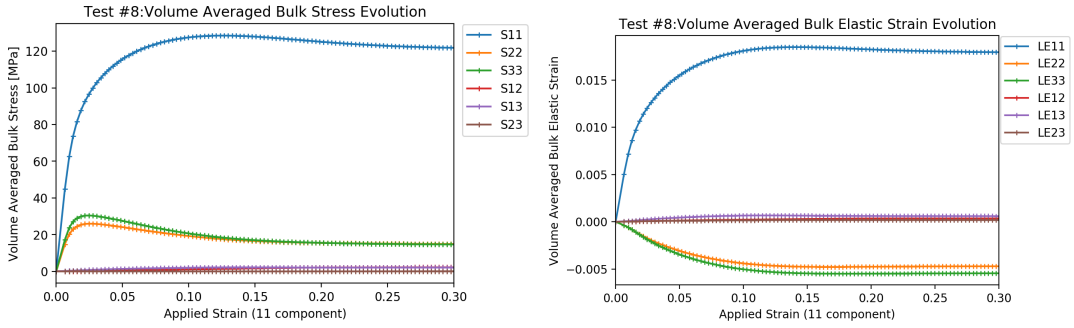


Figure 6.3: Test #8 results: SDEG on all cohesive elements.



(a) Stress

(b) Logarithmic Strain

Figure 6.4: Volume averaged bulk elastic stress and strain for test #8. These averages are performed over all bulk elements, except those which have at least one node in the RVE boundary.

#8 are shown in Figures 6.2 and 6.3, respectively. For this test, the expected fracture behaviour would be a single crack surface, perpendicular to the applied load. Damage should nucleate at the fiber interfaces and then the individual cracks should coalesce to form a single crack. Instead, it can be observed that two crack surfaces form near the RVE boundaries perpendicular to the loads, with little to no influence of the fibers in the damage onset. This behaviour will also be observed in subsequent analyses and section §6.2 is reserved for a more detailed discussion of this aspect. Test #9 was carried out using the same geometry and mesh as test #8, but pulling in the other transverse direction to see if the relative position of the fibers had any influence on this behaviour.

Because of this behaviour, all bulk averages were performed ignoring the elements which have at least, one node in the RVE boundary to remove the influence of this artifact. For test #8, such averages are shown in Figure 6.4. When looking at the plotted curves, it can be observed that the stress component aligned with the applied load is the highest. The difference in magnitude between S22 and S33 relative to S11 has the same order of magnitude as the Poisson's ratios of the materials involved. The components S22 and S33 being positive makes sense, since the material needs to be stretched laterally in order for the lateral dimensions of the volume element to remain constant, as no lateral strain was applied. On the other hand, the results for the strain are a bit more puzzling. Since a similar behaviour is observed in other analyses, further discussion regarding this aspect is done at the end of this chapter.

However, a previous attempt, test #3, didn't exhibit this behaviour at the boundaries. Its geometry and mesh are shown in Figure 6.5. It uses a larger fiber diameter in conjunction with a smaller fiber volume fraction, V_f. The reason for that is that this test was initially only supposed to be used for algorithm test purposes. This geometry was also created prior to the mesh fixing procedure implementation. The need

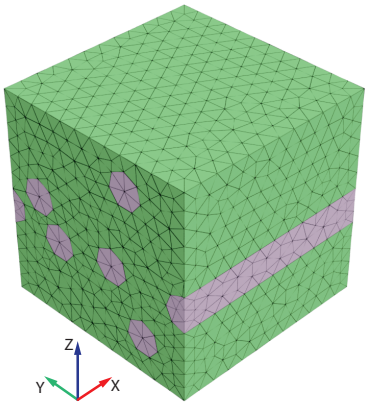


Figure 6.5: Geometry and mesh used in test #3, with a [90] layup.

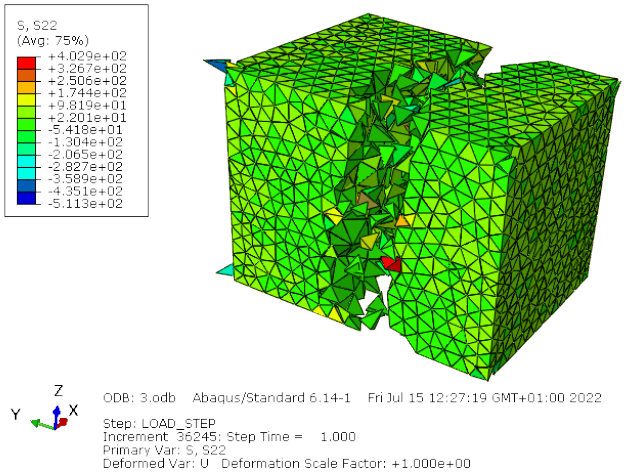


Figure 6.6: Test #3 results: S22 component of the stress tensor on all bulk elements.

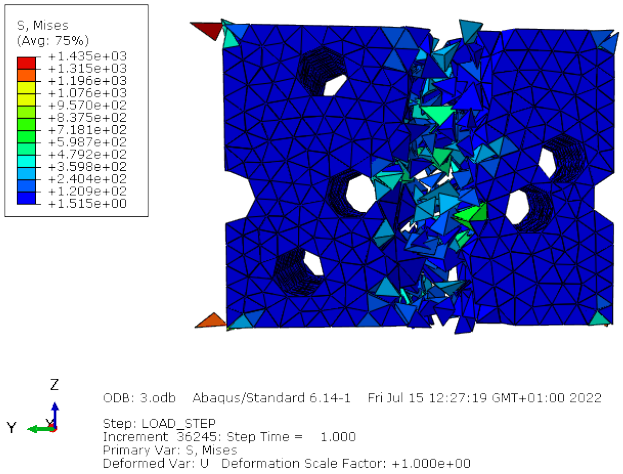
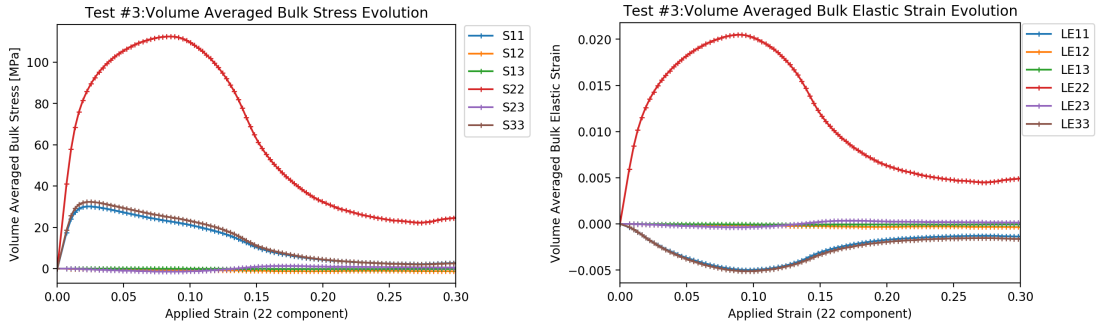


Figure 6.7: Test #3 results: Von Mises stress component on the matrix bulk elements.

to increase the V_f used, since the 15% used for this one is not large enough to be representative of typical composite materials used in aerospace applications, ended up motivating the creation of the mesh fixing step in the pre-processing workflow. This was because, in the absence of such step, the geometry creation and meshing steps had to be repeated until, purely by chance, a mesh which met the requirements was obtained. This was hard but achievable for low V_f values, but virtually impossible for larger values, requiring an extra step where this is fixed to be added.

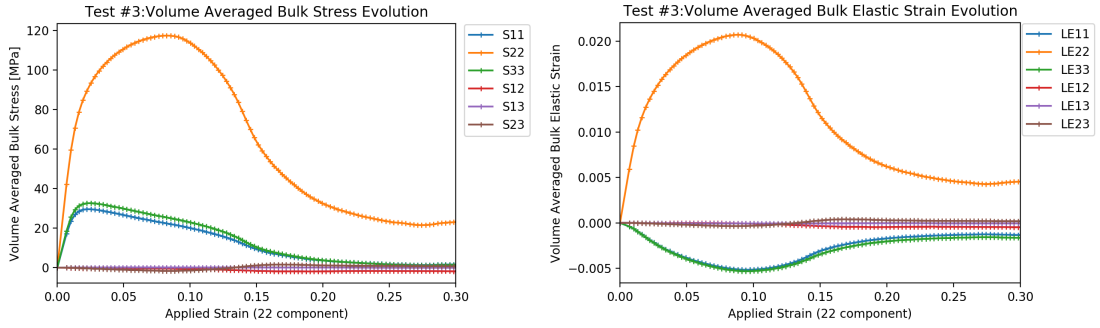
The contour plot for the stress component aligned with the applied load in test #3 is shown in Figure 6.6. To better understand the results, the contour plot with the Von Mises stress on the matrix's bulk elements



(a) Stress

(b) Logarithmic Strain

Figure 6.8: Volume averaged bulk elastic stress and strain for test #3. These averages are performed over all bulk elements.



(a) Stress

(b) Logarithmic Strain

Figure 6.9: Volume averaged bulk elastic stress and strain for test #3. These averages are performed over all bulk elements, except those which have at least one node in the RVE boundary.

can also be found in Figure 6.7. In the absence of the strange behaviour at the boundaries, the expected single crack perpendicular to the load vector is now observed. However, it does not start on the fiber matrix interfaces as expected, something that can be explained by the low V_f used. The fibers also remain intact due to the small load that they end up carrying. It can also be observed that, despite being fully cracked, some elements that are still loaded can still be founded, especially in the fragments within the cracked region. Some loaded elements can also be found near the corners. It is important to remember that the corner equations for PBCs are not actually used, since a node which is shared by three boundary faces is never present by the meshing software and, as a result, the corner nodes are actually treated as edge or face nodes.

The plots with the volume averaged bulk elastic stress and strain are shown in Fig. 6.8. The same plots but obtained while ignoring the elements with nodes on the RVE boundary are also shown in Figure 6.9 for consistency. Once again and as expected, the stress component aligned with the applied load is the highest. The same conclusions drawn before in terms of the curves' relative magnitudes also apply here. It can also be observed that there's a noticeable decrease in all stress curves due to the unloading caused by the crack formation. However, the S22 component doesn't go to zero as it should, as a result of the elements that are still loaded.

Longitudinal Loading

To test the fracture behaviour of unidirectional layups loaded along the fibers' direction, two tests were performed - test #12 and #13. Both of them use the same mesh and geometry as tests #8 and #9 (Fig. 6.1). The first one is identical to tests #8 and #9, with the only difference being the applied load. In the second one, the material properties used are slightly different, for reasons that will be explained later.

The contour plot for the stress component aligned with the applied load in test #12 is shown in Figure 6.10. For this load case, it was expected that fibers would break first, forming fracture surfaces perpendicular

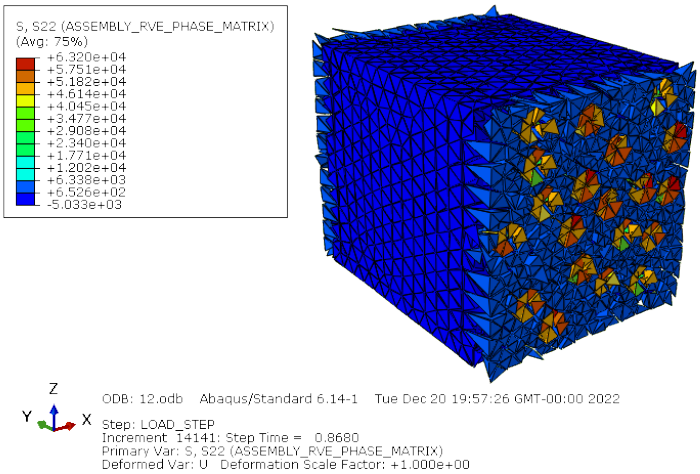
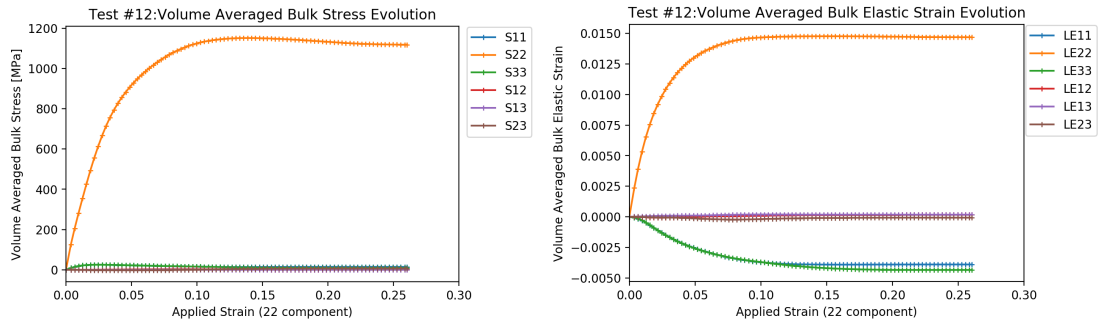


Figure 6.10: Test #12 results: S22 component of the stress tensor on all bulk elements.



(a) Stress

(b) Logarithmic Strain

Figure 6.11: Volume averaged bulk elastic stress and strain for test #12. These averages are performed over all bulk elements, except those which have at least one node in the RVE boundary.

to the applied load, which eventually link up through a matrix crack. Some fiber pull-out should also be observed. Unfortunately, this was not the case. As it can be seen, the same behaviour observed in tests #8 and #9 near the RVE boundaries is once again observed. Nonetheless, some differences can still be observed, such as the fact that the load is now carried almost entirely by the fibers due to the stiffness difference relative to the matrix.

Despite the unexpected behaviour near the boundaries, the fact that the expected fracture behaviour was not observed here can also be partially explained by another aspect. In an actual composite, it's the defects present in the fibers that serve as starting points for the cracks. Since the fibers are modelled as having no defects, there are no stress concentration points that can nucleate cracks. Local variations in the mesh can also do this, but they may not be enough. This is also in agreement with some tests performed using conditions similar to those of test #3, where this behaviour near the boundaries was only observed when pulling along the longitudinal direction but not along the transverse direction.

The plots with the volume averaged bulk elastic stress and strain are shown in Fig. 6.11. The obtained curves are very similar to those of Fig. 6.4, with only some small but still notable differences, such as that the stress components that are not aligned with the applied load, relative to the aligned one, are much smaller than before. The stress value at which the plateau is established in Fig. 6.4 is close to ultimate traction for matrix and interface (see Table 4.3). When the elements with nodes on the boundary are not ignored, these curves also exhibit the plateau at the same value and then an increase until the simulation ends. To check if it was possible that the ultimate traction for the fibers was excessively high and that the stress value at which they would break was never reached, a new test (#13) which uses different material properties was considered. For this test, the ultimate traction and fracture toughness were reduced by a factor of 5, as shown in Table 6.1.

Table 6.1: Material properties used for cohesive phases in analyses with reduced properties. The values which are different from those used in other analyses are marked in red.

Interface Type	E_{coh} [MPa]	t^o t_n [MPa]	t_s [MPa]	t_t [MPa]	G_f [kJ/mm^3]
Fiber - Fiber	1E+09	1000	1000	1000	1.5
Fiber - Matrix	1E+08	85	85	85	0.2
Matrix - Matrix	1E+08	80	80	80	0.2

For test #13, the contour plot for the stress component aligned with the applied load is shown in Fig. 6.12, being identical to that of Figure 6.10. The plots with the volume averaged bulk elastic stress and strain are shown in Fig. 6.13. As it can be seen, they also have the same shape as those shown in Fig. 6.11, although the stress value at which the plateau is established is now much smaller. As for the strain value, it is only slightly smaller.

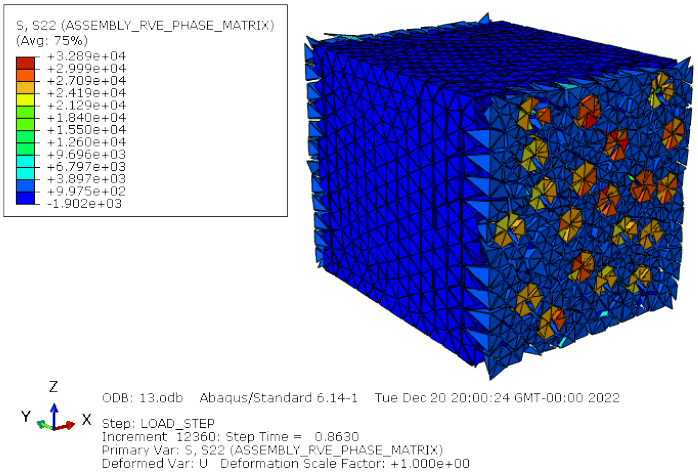
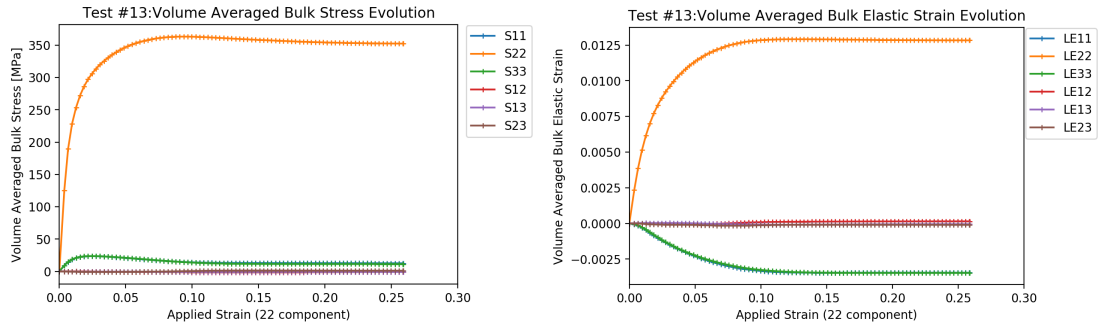


Figure 6.12: Test #13 results: S22 component of the stress tensor on all bulk elements.



(a) Stress

(b) Logarithmic Strain

Figure 6.13: Volume averaged bulk elastic stress and strain for test #13. These averages are performed over all bulk elements, except those which have at least one node in the RVE boundary.

Biaxial Loading

After testing the material's response when loading each direction individually, the next step is testing its response under combined loadings. For that, test #10 was carried out. This test uses the same mesh and geometry shown in Fig. 6.1 and is subjected to an in-plane biaxial stretching. Ideally, shear would have been applied, by stretching in one direction and compressing in the other. However, a contact algorithm is needed for that, since otherwise there will be interpenetration of bulk elements once fragments are formed.

For this test, it was expected that damage would start with a matrix crack at an angle. As this crack would then grow, some fiber-matrix debonding would be observed as well, leaving only the fibers to withstand the remaining load. The fibers eventually fail as well, but along crack planes perpendicular to the fiber's longitudinal direction, due to the way basal planes are arranged in carbon fibers.

The obtained contour plot for the stress component aligned with the applied load is shown in Fig. 6.14 Once again, the expected response was not observed. The same behaviour near the RVE boundaries is present, but this time in a particularly weird form. A total of 4 cracks are observed, one for each face where load is applied, which can also be easily identified when looking at the nodal displacements (Fig. 6.15). When looking at the stress distribution, another strange result can be seen: there are large differences in terms of the stress values between adjacent elements within the fibers.

As for the plots with the volume averaged bulk elastic stress and strain, these are shown in Fig. 6.16. The stress curves are very similar to those obtained for test #13, but the stress component in the transverse direction is larger, although still relatively small with the respect to longitudinal one due to the difference in stiffness of the phases present.

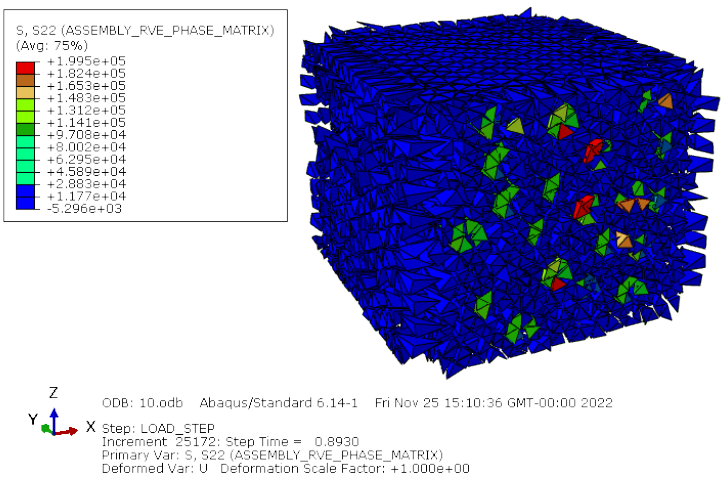
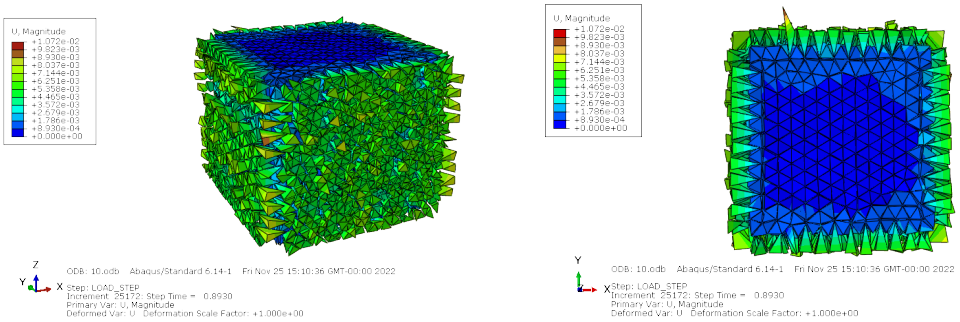
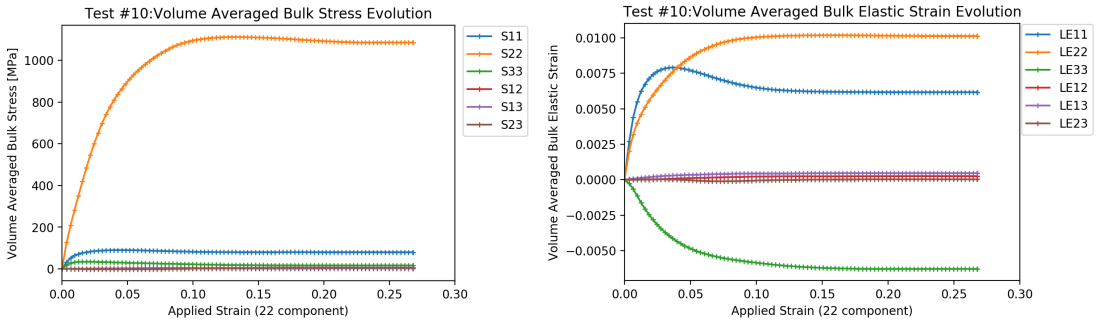


Figure 6.14: Test #10 results: S22 component of the stress tensor on all bulk elements.



(a) Default view (b) Top view

Figure 6.15: Test #10 results: magnitude of nodal displacements on all bulk elements.



(a) Stress (b) Logarithmic Strain

Figure 6.16: Volume averaged bulk elastic stress and strain for test #10. These averages are performed over all bulk elements, except those which have at least one node in the RVE boundary.

6.1.2. Multidirectional layups

Multidirectional layups are made up of multiple plies with different directions. With only two directions available, only a [0,90,0] layup was tested. The 90° ply has twice the thickness of a 0° ply, to ensure that the same volume is allocated to each fiber direction. This layup (Fig. 6.17) was loaded in the same way as it was done for unidirectional layups. The individual tests performed for each load case will be discussed next.

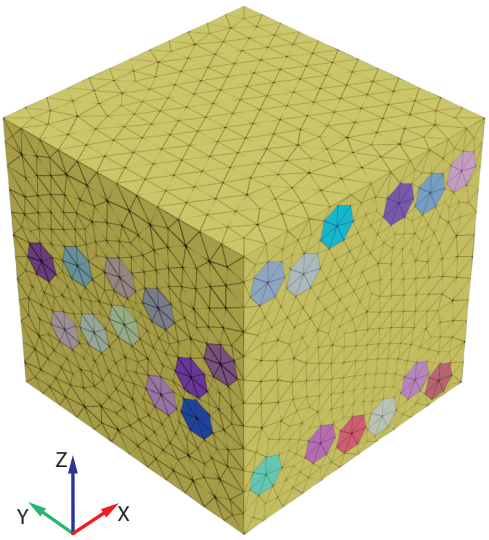


Figure 6.17: Geometry and mesh used in tests #1, #6, #11 and #14, with a [0,90,0] layup.

Longitudinal/Transverse Loading

Two tests - test #1 and #6 - were initially carried out where the layup was independently loaded along the two fiber directions, with test #1 being loaded in the Y direction and test #6 in X. Test #1 was also performed using the previously mentioned reduced properties (Table 6.1). This corresponds to test #14.

For a multidirectional layup made up of only 0° and 90° plies, the first plies to fail are those where fibers run perpendicular to the load. These are then followed by those where the fibers are aligned with the load. All these plies fail in the exact same way as they do when loaded individually.

For test #1, the obtained contour plot for the stress component aligned with the applied load is shown in

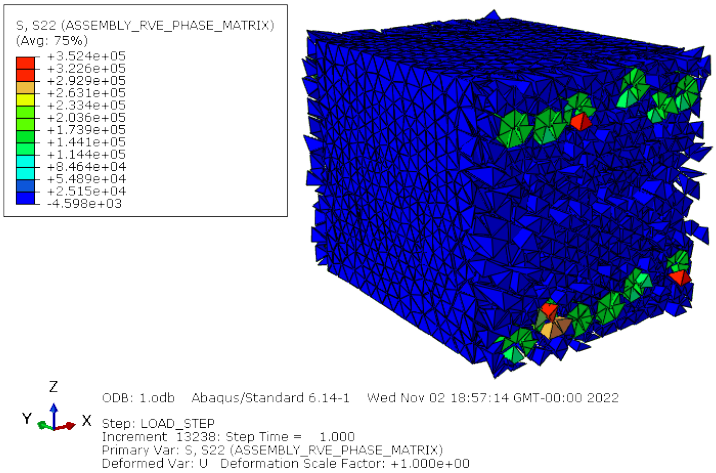
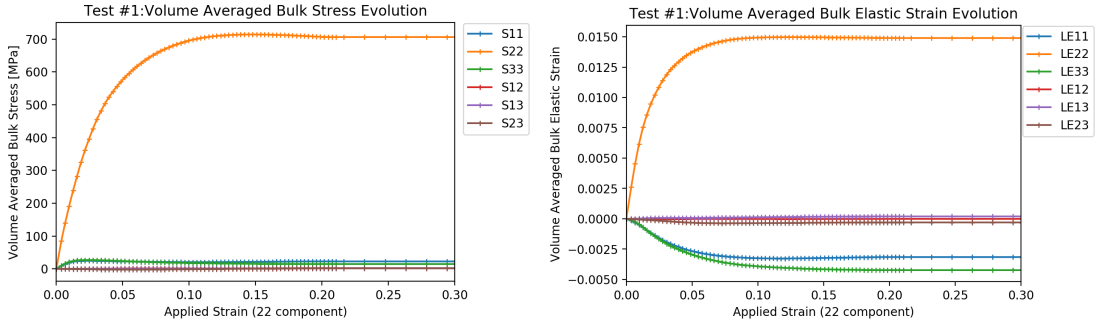


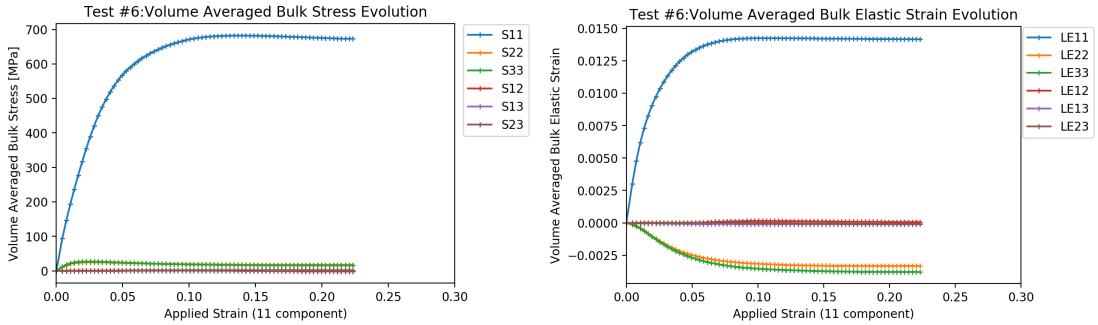
Figure 6.18: Test #1 results: S22 component of the stress tensor on all bulk elements.



(a) Stress

(b) Logarithmic Strain

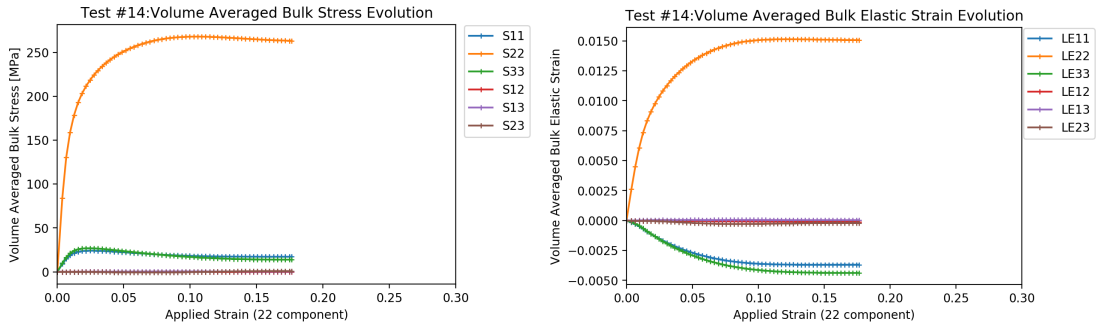
Figure 6.19: Volume averaged bulk elastic stress and strain for test #1. These averages are performed over all bulk elements, except those which have at least one node in the RVE boundary.



(a) Stress

(b) Logarithmic Strain

Figure 6.20: Volume averaged bulk elastic stress and strain for test #6. These averages are performed over all bulk elements, except those which have at least one node in the RVE boundary.



(a) Stress

(b) Logarithmic Strain

Figure 6.21: Volume averaged bulk elastic stress and strain for test #14. These averages are performed over all bulk elements, except those which have at least one node in the RVE boundary.

Fig. 6.14. Once again, the weird behaviour near the boundaries is observed. Despite this, it was possible to observe that damage started in the transversely loaded plies. The results obtained for test #6 are identical.

As for the plots with the volume averaged bulk elastic stress and strain, these are shown in Fig. 6.19 for test #1 and in Fig. 6.20 for test #6. Regarding the stress curves, they are very similar, apart from the switched components. Although there's loading in both directions, one stress component is much larger than the others due to the fibers that are aligned with the load. The peaks are roughly the same, which is a result of a similar amount of fibers in each direction. As for the strain curves, they are identical to those obtained for analyses done in unidirectional layups while applying the same strain tensor.

The same curves for test #14, with reduced properties, are shown in Fig. 6.21. Similarly to what was observed in the analyses with unidirectional layups, there are no significant changes, besides the lower plateau value.

Biaxial Loading

In terms of biaxial loading for multidirectional layups, only test #11 was performed. In this test, the previously mentioned mesh is subjected to an out-of-plane biaxial loading, i.e pulling along X and Z simultaneously. For this test, the expected fracture behaviour is a combination of what is expected for uniaxial loadings, but with some delamination cracks forming in the process as well.

The obtained contour plot for the stress component aligned with the applied load is shown in Fig. 6.22. Similarly to what was observed in other tests, the fracture behaviour didn't match what was expected and the weird behaviour at the boundaries was even especially pronounced in this case, probably as a result of some fibers being very close to the boundary.

The plots with the volume averaged bulk elastic stress and strain are shown in Fig. 6.23. These curves are

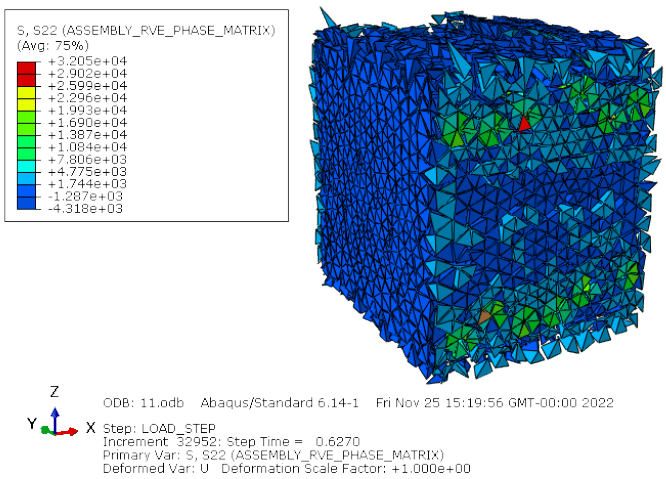
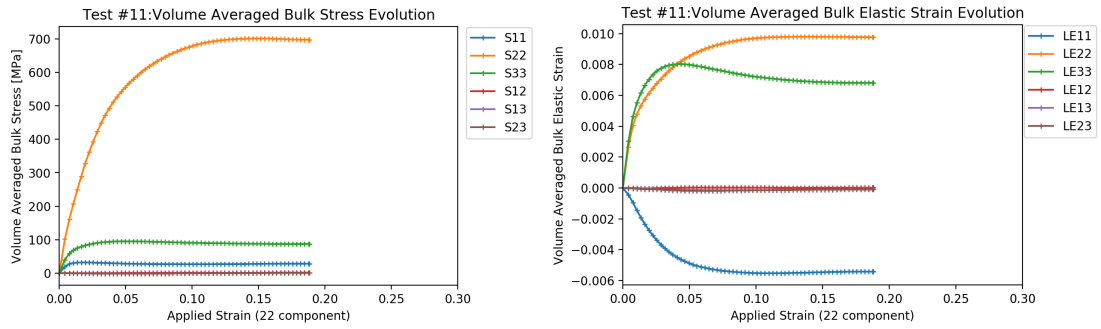


Figure 6.22: Test #11 results: S22 component of the stress tensor on all bulk elements.



(a) Stress

(b) Logarithmic Strain

Figure 6.23: Volume averaged bulk elastic stress and strain for test #11. These averages are performed over all bulk elements, except those which have at least one node in the RVE boundary.

nearly identical to test #10 (Fig. 6.16), except for a lower stress plateau, due to not all fibers being aligned with one of the directions where it is being pulled.

6.2. Final Remarks

So far, all analyses (except for test #3) exhibited a strange behaviour near the boundaries of the RVE where load was applied. In them, crack planes would form in the immediate vicinity of the loaded faces, regardless of layup and load direction.

Since it occurs near the boundaries where load is applied, an immediately considered possibility is that such errors could be caused by a wrong implementation of the PBCs or of the load application. For this reason, the way PBCs were enforced was checked more carefully. For that, dedicated scripts were created to check that everything was correct, before and after the analysis took place. The following aspects regarding the implementation of the equations were verified in the input file :

- All nodes which belong to boundary faces are used in, at least, one equation
- No nodes (dummy nodes not included) are subjected to edge and face equations simultaneously.
- No nodes (dummy nodes not included) are used in multiple equations.
- For each equation, the nodes in the associated pair are at the correct distance from each other.

All of these conditions were met. Checks were also made on the output. For all node pairs, the evolution of the difference in displacement between both nodes of a pair, with the dummy node contribution taken into account, was compared. The largest errors observed were in the order 10^{-10}, which is simply the result of round-off errors and has no impact on an analysis. With this, it was concluded that all equations were correctly applied in their current form. However, this doesn't exclude the possibility that some error might lie in the method by which the PBCs were implemented.

Nonetheless, other factors can also be a possible explanation for this behavior. One of them is that it might have something to do with the parameters used to specify the cohesive elements' response. On its own, this is not enough to explain the problem, but needs to be addressed since, even in test #3 where this behaviour wasn't present, the full unloading was not observed after the volume element fully cracked.

Another possibility is that it might be a mesh-related issue. First, the mesh size might be too large relative to the phases involved. In almost all analyses performed, there were only 2 or 3 elements in any direction perpendicular to the fiber's axis. Such a low number of elements is insufficient to fully capture the stress state within a fiber. Since the same mesh size was used everywhere, the same comments are valid for any point in the matrix as well. This hypothesis is supported by the fact that a larger fiber diameter was used in test #3. However, any analysis able to verify it is currently impossible to carry out due to the prohibitively high computational cost. It might also be related to a small bias introduced by the mesh fixing procedure. When comparing cases in different edges, if there's a conflict, case 2 is always favoured instead of case 1. Since this isn't a rare occurrence, there's a predominance of edge segments associated with case 2. This leads to a lot of cohesive elements close to each other with an almost identical orientation, which might favour the

formation of a crack there. This is supported by the fact that the only test presented here where the behaviour was not observed is also the only one where the mesh didn't go through the mesh fixing procedure. Moreover, the presence of fibers crossing the boundaries might also prevent this, by inserting cohesive elements with a larger ultimate traction. Once again, this is not enough to explain the behaviour, since it was also observed for initial tests similar to test #3 when load was applied along the fibers' direction.

In order to test all of these possible causes, many analyses would need to be carried out, something which is impossible to do in this thesis alone given the time each analysis takes. As such, this is left as an open problem for future work on this topic.

Besides the problem with the boundaries, there's also a problem associated with the curves for the volume averaged logarithmic strain. As seen before, the average bulk strain and the fracture strain correspond to the applied strain tensor when combined (eq. 3.10). In all tests presented, there's a mismatch. This can be explained by the missing crack's contribution. However, that's not the only problem with the strain curves. While the stress-aligned component matches its strain counterpart, the same cannot be said by the others. In all tests presented, the strain components for directions where load is not applied are negative, while their strain counterparts are positive. The stress curves' sign can be understood as being a result of the stretching required to ensure that the volume element doesn't contract laterally, as enforced by the applied strain tensor. As for the strain curves, it was expected that they would have the same sign as the stress curves One possible explanation is that the fracture strain compensates for this such that their sum is equal to zero at any point in time. However, even at the start of the analysis, where there's almost no damage, the curves are already negative. This problem can also be a windfall from the problem with PBCs, since it is related to how the applied strain tensor is enforced. As such, no definitive conclusions as to why this occurs could be drawn from the available data.

Lastly, even if both of these problems were not present, these results can only be considered as valid after a full verification campaign is completed. This entails a mesh convergence study, a volume element size convergence and a statistical convergence regarding the distribution of fibers within the volume element. It is also necessary to check if Hill-Mandel's condition, particularly the crack-based one, is also verified.

7

Conclusions and Recommendations

7.1. Conclusions

Over the last years, there has been a growing presence of composite materials in aerospace applications. This adoption has so far been hindered by the still limited understanding of how damage nucleates and grows in these materials. This is a result of the complexity associated with such behaviour, which is governed by the microsctructures of such materials and can be very sensitive to small changes. Multiscale methods, and more specifically a methodology currently being developed at TU Delft, can provide a way of modelling fracture at macroscopic scales, while implicitly taking into account the influence on the material's response of all phases present in its microstructure. Such information is passed to the larger scales in the form of effective traction separation laws (ETSLs), which dictate the response of the associated effective macroscopic crack, with an area and orientation that is also derived from the results obtained at the microscopic scales.

In this thesis, this newly developed multiscale methodology is extended to 3D. This is an important step towards a more versatile way of fracture in composite materials. With it, the plane strain assumption made for 2D problems is removed and it becomes possible to study problems with multiple fiber directions besides 0° and 90°.

In terms of its analytical formulation, the 2D and 3D formulations are nearly identical. Most of the changes required for its extension are simply a change in the dimension of the domain where integration is performed. Besides that, some relations and concepts, such as those which rely on angles, were also adapted to 3D.

On the other hand, in order to accommodate 3D problems, significant changes had to be made in the Python scripts used to numerically implement this methodology. Regarding the geometry generation step, the changes implemented were successful at creating a tool capable of generating multiple-ply layups with random spacial distributions of straight fibers within each ply. The respective periodic replicas of fibers that cross the RVE boundary are also created, as required. Due to this periodicity requirement, only certain angles are allowed to avoid infinite periodic replicas. The geometry is then meshed in Gmsh. Limitations related to this software prevented the use of orientations other than 0° and 90° in combination with the required periodicity. Additionally, due to how PBCs are enforced, an extra step where the generated meshes are fixed was added, since the specific requirements for a proper implementation of PBCs could not be enforced using the available Gmsh controls. For those meshes which met all those requirements, cohesive elements are then added between all bulk elements and the INP file describing the Abaqus model is created.

Regarding the post-processing, the required changes were not fully implemented and/or tested due to time constraints and a lack of good data. Nonetheless, the algorithms which were implemented but could not be tested were described and suggestions were given on possible ways to implement the remaining steps.

With the required changes in place, some analyses were performed. Only a relatively small number of analyses were carried out due to the high computational cost associated with each one. These analyses focused mainly on representative load cases for both unidirectional and multidirectional layups. A verification campaign was also not performed for the same reasons. As for the results, they were not what was expected. With the exception of a single analysis, all others displayed a strange behaviour near the RVE boundaries where load was applied. Even in the presence of such behaviour, some physically relevant aspects could be observed in the results, but they were mostly overshadowed by this problem. Possible reasons as to why this

might occur are laid out, but definitive conclusions could not be drawn from the available data and more analyses to test some hypotheses could not be carried out due to time constraints.

7.2. Recommendations

Taking into consideration what was achieved and observed during this thesis project, as well as the overall research objectives, the following recommendations are made regarding future work on this topic:

- Evaluate the possible causes behind the problem at the boundaries. During this thesis, a strange behaviour near the RVE boundaries where load was applied was observed, which had a significant impact on the results. Some hypotheses as to why this was observed were presented already. In order to properly assess its causes, as well as implement the required changes, time and resources need to be dedicated to tackling this issue. Solving it is key to unlock the expected advantages of 3D over 2D simulations. Without addressing this issue first, it is also pointless to pursue any other recommendation related to 3D, as the results are of insufficient quality for any relevant conclusions to be drawn from them.

- Complete the post-processing in 3D. During this thesis, it was also mentioned that it was not possible to implement and test the post-processing segment. Ideally, this should only be pursued once the strange behaviour near the boundaries has been fixed.

- Implement a contact algorithm. This is a key step in order to unlock the full spectrum of load cases, as shear and compression cannot be handled properly in the software's current form. Even for load cases where no compression loads are applied in any direction, some local interpenetration of elements can still be observed. Due to its complexity and the high number of analyses required to test and verify this modification, it might be better to first implement it in 2D and then adapt the algorithm to 3D.

- Include plasticity in the model. This is relatively simple to implement, since it requires minimal modifications in the current scripts. It could also be implemented in 2D before 3D to facilitate the verification procedure.

- Use more accurate material models. Besides the already mentioned plasticity, the results could be made more realistic by including defects on fibers and by forcing through the mesh that fiber cracks only form along planes perpendicular to their axial direction. Fibers can also be slightly curved with relative ease in the current implementation, although this only becomes significant under compression loads.

Bibliography

[1] T. L. Anderson. *Fracture mechanics: Fundamentals and Applications*. CRC Press, 2005. ISBN 9781420058215.

[2] I. Babuška, G. Caloz, and J. E. Osborn. Special finite element methods for a class of second order elliptic problems with rough coefficients. *SIAM Journal on Numerical Analysis*, 31(4):945–981, 1994. URL `http://www.jstor.org/stable/2158110`.

[3] G. I. Barenblatt. The mathematical theory of equilibrium of crack in brittle fracture. *Advances in Applied Mechanics*, 7:55–129, 1962.

[4] T. Belytschko and T. Black. Elastic crack growth in finite elements withminimal remeshing. *International Journal for Numerical Methods in Engineering*, 45:601–620, 1999.

[5] V. A. Buryachenko, N. J. Pagano, R. Y. Kim, and J. E. Spowart. Quantitative description and numerical simulation of random microstructures of composites and their effective elastic moduli. *International Journal of Solids and Structures*, 40(1):47–72, 2003. doi: 10.1016/S0020-7683(02)00462-6.

[6] G. T. Camacho and M. Ortiz. Computational modelling of impact damage in brittle materials. *International Journal of Solids and Structures*, 33(20):2899–2938, 1996. doi: 10.1016/0020-7683(95)00255-3.

[7] P. Camanho and C. Dávila. Mixed-mode decohesion finite elements for the simulation of delamination in composite materials. *NASA/TM-2002-211737*, 2002.

[8] P. P. Camanho, C. G. Davila, and M. F. de Moura. Numerical simulation of mixed-mode progressive delamination in composite materials. *Journal of Composite Materials*, 37(16):1415–1438, 2003. doi: 10.1177/0021998303034505.

[9] J. L. Chaboche, F. Feyel, and Y. Monerie. Interface debonding models: a viscous regularization with a limited rate dependency. *International Journal of Solids and Structures*, 38(18):3127–3160, 2001. doi: 10.1016/S0020-7683(00)00053-6.

[10] C. A. Coulomb. *Essai sur une application des regles de maximis & minimis à quelques problèmes de statique, relatifs à l'architectur*. De l'Imprimerie Royale, Paris, 1776.

[11] I. M. Daniel and O. Ishai. *Engineering mechanics of composite materials*. Oxford University Press, 2004. ISBN 9780195150971.

[12] Dassault Systèmes Simulia Corp. *Abaqus 6.12 Analysis User's Manual: 32.5.6 Defining the constitutive response of cohesive elements using a traction-separation description*, . URL `http://abaqus.software.polimi.it/v6.12/books/usb/default.htm?startat=pt06ch32s05alm44.html`.

[13] Dassault Systèmes Simulia Corp. *Abaqus 6.12 Analysis User's Manual: 29.5.6 Defining the constitutive response of cohesive elements using a traction-separation description*, . URL `http://130.149.89.49:2080/v6.10/books/usb/default.htm?startat=pt06ch29s05alm45.html#usb-elm-ecohesivebehavior-regularize`.

[14] Dassault Systèmes Simulia Corp. *Abaqus 6.12 Analysis User's Manual: 28.1.4 Three-dimensional solid element library*, . URL `http://130.149.89.49:2080/v2016/books/usb/default.htm?startat=pt06ch28s01ael03.html`.

[15] Dassault Systèmes Simulia Corp. *Abaqus 6.12 Analysis User's Manual: 32.5.4 Defining the cohesive element's initial geometry*, . URL `http://130.149.89.49:2080/v2016/books/usb/default.htm?startat=pt06ch32s05alm43.html`.

[16] Dassault Systèmes Simulia Corp. *Abaqus 6.12 Analysis User's Manual: 35.2.1 Linear constraint equations*, . URL `http://130.149.89.49:2080/v2016/books/usb/default.htm?startat=pt06ch32s05alm43.html`.

[17] R. de Borst, M. A. Crisfield, J. J. C. Remmers, and C. V. Verhoosel. *Non-Linear Finite Element Analysis of Solids and Structures*. John Wiley & Sons, 2012. ISBN 9780470666449.

[18] L. A. de Oliveira and M. V. Donadon. Delamination analysis using cohesive zone model: A discussion on traction-separation law and mixed-mode criteria. *Engineering Fracture Mechanics*, 228:106922, 2020. ISSN 0013-7944. doi: https://doi.org/10.1016/j.engfracmech.2020.106922.

[19] B. C. Do, W. Liu, Q. D. Yang, and X. Y. Su. Improved cohesive stress integration schemes for cohesive zone elements. *Engineering Fracture Mechanics*, 107:14–28, 2013. doi: 10.1016/j.engfracmech.2013.04.009.

[20] D. C. Drucker and W. Prager. Soil mechanics and plastic analysis or limit design. *Quarterly of Applied Mathematics*, 10(2):157–165, 1952. doi: 10.1090/qam/48291.

[21] W. J. Drugan and J. R. Willis. A micromechanics-based nonlocal constitutive equation and estimates of representative volume element size for elastic composites. *Journal of the Mechanics and Physics of Solids*, 44(4):497–524, 1996. doi: 10.1016/0022-5096(96)00007-5.

[22] D. Dugdale. Yielding of steel sheets containing slits. *Journal of the Mechanics and Physics of Solids*, 8: 100–108, 1960.

[23] M. Elices, G. V. Guinea, J. Gomez, and J. Planas. The cohesive zone model: advantages, limitations and challenges. *Engineering Fracture Mechanics*, 69(2):137–163, 2002. ISSN 0013-7944. doi: 10.1016/S0013-7944(01)00083-2.

[24] J. Fan and E. B. Tadmor. Rescaling cohesive element properties for mesh independent fracture simulations. *Engineering Fracture Mechanics*, 213:89–99, 2019. doi: 10.1016/j.engfracmech.2019.03.035.

[25] Frédéric Feyel. A multilevel finite element method (fe2) to describe the response of highly non-linear structures using generalized continua. *Computer Methods in Applied Mechanics and Engineering*, 192 (28):3233–3244, 2003. doi: 10.1016/S0045-7825(03)00348-7.

[26] Frédéric Feyel and Jean-Louis Chaboche. Fe2 multiscale approach for modelling the elastoviscoplastic behaviour of long fibre sic/ti composite materials. *Computer Methods in Applied Mechanics and Engineering*, 183(3):309–330, 2000. doi: 10.1016/S0045-7825(99)00224-8.

[27] D. François, A. Pineau, and A. Zaoui. *Mechanical Behaviour of Materials Volume II: Fracture Mechanics and Damage*. Springer, 2013. ISBN 9789400749290.

[28] Y. F. Gao and A. F. Bower. A simple technique for avoiding convergence problems in finite element simulations of crack nucleation and growth on cohesive interfaces. *Modelling and Simulation in Materials Science and Engineering*, 12(3):453–463, 2004. doi: 10.1088/0965-0393/12/3/007.

[29] M. G. D. Geers, V. Kouznetsova, and W. A. M. Brekelmans. Multi-scale computational homogenization: Trends and challenges. *Journal of Computational and Applied Mathematics*, 234(7):2175–2182, 2010. doi: 10.1016/j.cam.2009.08.077.

[30] M. G. D. Geers, V. G. Kouznetsova, K. Matouš, and J. Yvonnet. Homogenization methods and multiscale modeling: Nonlinear problems. In E. Stein, R. Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics*, pages 1–34. John Wiley & Sons, 2nd edition, 2017. ISBN 9781119176817. doi: 10.1002/9781119176817.ecm2107.

[31] C. Geuzaine and J. Remacle. *Gmsh 4.9.3*, 2022. URL `https://gmsh.info/doc/texinfo/gmsh.html`.

[32] I. M. Gitman, H. Askes, and L. J. Sluys. Representative volume: Existence and size determination. *Engineering Fracture Mechanics*, 74(16):2518–2534, 2007. doi: 10.1016/j.engfracmech.2006.12.021.

[33] V. Giurgiutiu. *Structural Health Monitoring of Aerospace Composites*. Academic Press, 2016. ISBN 978-0-12-409605-9. doi: 10.1016/C2012-0-07213-4.

[34] C.s González and J. LLorca. Mechanical behavior of unidirectional fiber-reinforced polymers under transverse compression: Microscopic mechanisms and modeling. *Composites Science and Technology*, 67(13):2795–2806, 2007. doi: 10.1016/j.compscitech.2007.02.001.

[35] A. A. Griffith. The phenomena of rupture and flow in solids. In *Philosophical Transactions*, volume 221 of *Series A*, pages 163–198, 1920.

[36] D. Gross and T. Seelig. *Fracture Mechanics: With an Introduction to Micromechanics*. Springer, 2018. ISBN 9783319710891.

[37] F. Guo, P. Huang, Y. Li, N. Hu, and S. Fu. Multiscale modeling of mechanical behaviors of carbon fiber reinforced epoxy composites subjected to hygrothermal aging. *Composite Structures*, 256:113098, 2021. doi: 10.1016/j.compstruct.2020.113098.

[38] R. Gutkin, S. T. Pinho, P. Robinson, and P. T. Curtis. On the transition from shear-driven fibre compressive failure to fibre kinking in notched cfrp laminates under longitudinal compression. *Composites Science and Technology*, 70(8):1223–1231, 2010. doi: 10.1016/j.compscitech.2010.03.010.

[39] L. Hamitouche, M. Tarfaoui, and A. Vautrin. An interface debonding law subject to viscous regularization for avoiding instability: Application to the delamination problems. *Engineering Fracture Mechanics*, 75(10):3084–3100, 2008. doi: 10.1016/j.engfracmech.2007.12.014.

[40] C. He, J. Ge, B. Zhang, J. Gao, S. Zhong, W. K. Liu, and D. Fang. A hierarchical multiscale model for the elastic-plastic damage behavior of 3d braided composites at high temperature. *Composites Science and Technology*, 196:108230, 2020. doi: 10.1016/j.compscitech.2020.108230.

[41] M. J. Hinton, A. S. Kaddour, and P. D. Soden. Chapter 6.1 - predictive capabilities of nineteen failure theories and design methodologies for polymer composite laminates. part b: Comparison with experiments. In *Failure Criteria in Fibre-Reinforced-Polymer Composites*, pages 1073–1221. Elsevier, 2004. ISBN 9780080444758. doi: 10.1016/B978-008044475-8/50038-X.

[42] M. Hojo, M. Mizuno, T. Hobbiebrunken, T. Adachi, M. Tanaka, and S. K. Ha. Effect of fiber array irregularities on microscopic interfacial normal stress states of transversely loaded ud-cfrp from viewpoint of failure initiation. *Composites Science and Technology*, 69(11):1726–1734, 2009. doi: 10.1016/j.compscitech.2008.08.032.

[43] IATA. Iata 20-year air passenger forecast, 2021. URL `https://www.iata.org/contentassets/fe5b20e8aae147c290fc4880f120c969/4679_passenger-forecast-infographic-update_v2.pdf`. (Accessed 20 January 2022).

[44] C. E. Inglis. Stresses in a plate due to the presence of cracks and sharp corners. *Transactions of the Institute of Naval Architects*, 55:219–241, 1913.

[45] G. R. Irwin. Onset of fast crack propagation in high strength steel and aluminum alloys. In *Sagamore Research Conference Proceedings*, volume 2, page 289–305, 1956.

[46] H. Jiang and Y. Xie. A note on the mohr–coulomb and drucker–prager strength criteria. *Mechanics Research Communications*, 38(4):309–314, 2011. doi: 10.1016/j.mechrescom.2011.04.001.

[47] H. Jiang and Y. Xie. Corrigendum to "a note on the mohr–coulomb and drucker–prager strength criteria" [mech. res. commun. 38 (2011) 309–314]. *Mechanics Research Communications*, 64:57, 2015. doi: 10.1016/j.mechrescom.2015.01.001.

[48] P. Johnson and F-K Chang. Characterization of matrix crack-induced laminate failure—part i: experiments. *Journal of Composite Materials*, 35(22):2009–2035, 2001. doi: 10.1106/7RN1-PFBN-XQR9-3KDK.

[49] P. Johnson and F-K Chang. Characterization of matrix crack-induced laminate failure—part ii: analysis and verifications. *Journal of Composite Materials*, 35(22):2037–2074, 2001. doi: 10.1106/VTQ8-TXDY-72Q6-677V.

[50] A. J. Kinloch and R. J. Young. *Fracture Behaviour of Polymers*. Springer Netherlands, 1995. ISBN 9789401715966. doi: 10.1007/978-94-017-1594-2.

[51] V. Kouznetsova. *Computational homogenization for the multi-scale analysis of multi-phase materials*. PhD thesis, Technische Universiteit Eindhoven, 2002.

[52] V. Kouznetsova, M. G. D. Geers, and W. A. M. Brekelmans. Multi-scale constitutive modelling of heterogeneous materials with a gradient-enhanced computational homogenization scheme. *International Journal for Numerical Methods in Engineering*, 54(8):1235–1260, 2002. doi: 10.1002/nme.541.

[53] J. Llorca, C. González, and J. Segurado. 5 - finite element and homogenization modelling of materials. In Z. Xiao Guo, editor, *Multiscale Materials Modelling*, pages 121–147. Woodhead Publishing, 2007. ISBN 9781845690717. doi: 10.1533/9781845693374.121.

[54] J. LLorca, C. González, J. M. Molina-Aldareguía, J. Segurado, R. Seltzer, F. Sket, M. Rodríguez, S. Sádaba, R. Muñoz, and L. P. Canal. Multiscale modeling of composite materials: a roadmap towards virtual testing. *Advanced Materials*, 23(44):5130–5147, 2011. doi: 10.1002/adma.201101683.

[55] S. Loehnert and T. Belytschko. A multiscale projection method for macro/microcrack simulations. *International Journal for Numerical Methods in Engineering*, 71(12):1466–1482, 2007. doi: 10.1002/nme.2001.

[56] X. Lu, B. Y. Chen, V. B. C. Tan, and T. E. Tay. A separable cohesive element for modelling coupled failure in laminated composite materials. *Composites Part A: Applied Science and Manufacturing*, 107:387–398, 2018. doi: 10.1016/j.compositesa.2018.01.014.

[57] P. K. Mallick. *Fiber-reinforced composites: Materials, manufacturing, and design, third edition*. CRC Press, 2007. ISBN 9780849342059.

[58] C. McCarthy and T. Vaughan. 14 - micromechanical failure analysis of advanced composite materials. In P. P. Camanho, S. R. HallettP. P. Camanho, and S. R. Hallett, editors, *Numerical Modelling of Failure in Advanced Composite Materials*, pages 379–409. Woodhead Publishing, 2015. doi: 10.1016/B978-0-08-100332-9.00014-1.

[59] 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(1):289–314, 1996. doi: 10.1016/S0045-7825(96)01087-0.

[60] J. M. Melenk and I. Babuška. The partition of unity method. *International Journal for Numerical Methods in Engineering*, 40(4):727–758, 1997. doi: 10.1002/(SICI)1097-0207(19970228)40:4<727::AID-NME86>3.0.CO;2-N.

[61] A. R. Melro, P. P. Camanho, and S. T. Pinho. Influence of geometrical parameters on the elastic response of unidirectional composite materials. *Composite Structures*, 94(11):3223–3231, 2012. doi: 10.1016/j.compstruct.2012.05.004.

[62] A. R. Melro, P. P. Camanho, F. M. Andrade Pires, and S. T. Pinho. Micromechanical analysis of polymer composites reinforced by unidirectional fibres: Part ii – micromechanical analyses. *International Journal of Solids and Structures*, 50(11):1906–1915, 2013. doi: 10.1016/j.ijsolstr.2013.02.007.

[63] J. Mergheim, E. Kuhl, and P. Steinmann. A hybrid discontinuous galerkin/interface method for thecomputational modelling of failure. *Communications in Numerical Methods in Engineering*, 20(7):511–519, 2004. doi: 10.1002/cnm.689.

[64] R. E. Miller and E. B. Tadmor. A unified framework and performance benchmark of fourteen multiscale atomistic/continuum coupling methods. *Modelling and Simulation in Materials Science and Engineering*, 17(5):053001, 2009. doi: 10.1088/0965-0393/17/5/053001.

[65] S. Mohammadi. *Extended finite element method for fracture analysis of structures*. Oxford: Blackwell Pub, 2008. ISBN 9781405170604.

[66] O. Mohr. *Abhandlungen aus dem Gebiete der technischen Mechanik*. W. Ernst & Sohn, Berlin, 1914.

[67] J. I. Múgica, C. S. Lopes, F. Naya, M. Herráez, V. Martínez, and C. González. Multiscale modelling of thermoplastic woven fabric composites: From micromechanics to mesomechanics. *Composite Structures*, 228:111340, 2019. doi: 10.1016/j.compstruct.2019.111340.

[68] F. Naya, C. S. Lopes, C. González, and J. LLorca. 15 - computational micromechanics strategies for the analysis of failure in unidirectional composites. In P. P. Camanho and S. R. Hallett, editors, *Numerical Modelling of Failure in Advanced Composite Materials*. Woodhead Publishing, 2015. ISBN 978-0-08-100332-9. doi: 10.1016/B978-0-08-100332-9.00015-3.

[69] F. Naya Montáns. *Prediction of mechanical properties of unidirectional FRP plies at different environmental conditions by means of computational micromechanics*. PhD thesis, Universidad Politécnica de Madrid, 2017.

[70] A. Needleman. A continuum model for void nucleation by inclusion debonding. *Journal of Applied Mechanics*, 54(3):525–531, September 1987. ISSN 0021-8936. doi: 10.1115/1.3173064.

[71] V. D. Nguyen, E. Béchet, C. Geuzaine, and L. Noels. Imposing periodic boundary condition on arbitrary meshes by polynomial interpolation. *Computational Materials Science*, 55:390–406, 2011. doi: 10.1016/j.commatsci.2011.10.017.

[72] V. D. Nguyen, G. Becker, and L. Noels. Multiscale computational homogenization methods with a gradient enhanced scheme based on the discontinuous galerkin formulation. *Computer Methods in Applied Mechanics and Engineering*, 260:63–77, 2013. doi: 10.1016/j.cma.2013.03.024.

[73] V. P. Nguyen. Discontinuous galerkin/extrinsic cohesive zone modeling: Implementation caveats and applications in computational fracture mechanics. *Engineering Fracture Mechanics*, 128:37–68, 2014. doi: 10.1016/j.engfracmech.2014.07.003.

[74] V. P. Nguyen. An open source program to generate zero-thickness cohesive interface elements. *Advances in Engineering Software*, 74:27–39, 2014. doi: 10.1016/j.advengsoft.2014.04.002.

[75] K. Park and G. Paulino. Cohesive zone models: A critical review of traction-separation relationships across fracture surfaces. *Applied Mechanics Reviews*, 64:1002–, 2011. doi: 10.1115/1.4023110.

[76] K. Park, H. Choi, and G. H. Paulino. Assessment of cohesive traction-separation relationships in abaqus: A comparative study. *Mechanics Research Communications*, 78:71–78, 2016. doi: 10.1016/j.mechrescom.2016.09.004.

[77] B. Patzák and M. Jirásek. Adaptive resolution of localized damage in quasi-brittle materials. *Journal of Engineering Mechanics ASCE*, 130(6):720–732, 2004.

[78] S. A. Ponnusami, J. Krishnasamy, S. Turteltaub, and S. van der Zwaag. Elucidating the effect of cohesive zone length in fracture simulations of particulate composites. *Engineering Fracture Mechanics*, 268: 108431, 2022. doi: 10.1016/j.engfracmech.2022.108431.

[79] J. Remmers, R. Borst, and A. Needleman. A cohesive segments method for the simulation of crack growth. *Computational Mechanics*, 31:69–77, 2003. doi: 10.1007/s00466-002-0394-z.

[80] J. R. Rice. A path independent integral and the approximate analysis of strain concentration by notches and cracks. *Journal of Applied Mechanics*, 35:379–386, 1968.

[81] M. Romanowicz. A numerical approach for predicting the failure locus of fiber reinforced composites under combined transverse compression and axial tension. *Computational Materials Science*, 51(1): 7–12, 2012. doi: 10.1016/j.commatsci.2011.07.039.

[82] R. Russo and B. Chen. Overcoming the cohesive zone limit in composites delamination: modeling with slender structural elements and higher-order adaptive integration. *International Journal for Numerical Methods in Engineering*, 121:5511– 5545, 2020. doi: 10.1002/nme.6497.

[83] C. Sarrado, F. A. Leone, and A. Turon. Finite-thickness cohesive elements for modeling thick adhesives. *Engineering Fracture Mechanics*, 168:105–113, 2016. doi: 10.1016/j.engfracmech.2016.03.020.

[84] 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. doi: 10.1002/nme.1620360104.

[85] J. Schijve, editor. *Fatigue of Structures and Materials*. Springer Netherlands, 2009. ISBN 9781402068089. doi: 10.1007/978-1-4020-6808-9.

[86] R. M. Sencu, Z. Yang, Y. C. Wang, P. J. Withers, and C. Soutis. Multiscale image-based modelling of damage and fracture in carbon fibre reinforced polymer composites. *Composites Science and Technology*, 198:108243, 2020. doi: 10.1016/j.compscitech.2020.108243.

[87] A. Sharma, S. Daggumati, A. Gupta, and W. Van Paepegem. On the prediction of the bi-axial failure envelope of a ud cfrp composite lamina using computational micromechanics: Effect of microscale parameters on macroscale stress–strain behavior. *Composite Structures*, 251:112605, 2020. doi: 10.1016/j.compstruct.2020.112605.

[88] J. Shi. *Discrete Modelling of Plasticity and Martensitic Transformations*. PhD thesis, Shanghai Jiao Tong University, 2009.

[89] M. Silani, S. Ziaei-Rad, H. Talebi, and T. Rabczuk. A semi-concurrent multiscale approach for modeling damage in nanocomposites. *Theoretical and Applied Fracture Mechanics*, 74:30–38, 2014. doi: 10.1016/j.tafmec.2014.06.009.

[90] G. Soni, R. Singh, M. Mitra, and B. G. Falzon. Modelling matrix damage and fibre–matrix interfacial decohesion in composite laminates via a multi-fibre multi-layer representative volume element (m2rve). *International Journal of Solids and Structures*, 51(2):449–461, 2014. doi: 10.1016/j.ijsolstr.2013.10.018.

[91] D. W. Spring and G. H. Paulino. A growing library of three-dimensional cohesive elements for use in abaqus. *Engineering Fracture Mechanics*, 126:190–216, 2014. doi: 10.1016/j.engfracmech.2014.04.004.

[92] R. Suárez-Millán. Multiscale simulations for fracture prediction in composite materials: extension for anisotropic microstructures. Master's thesis, Delft University of Technology, 2018. URL `http://resolver.tudelft.nl/uuid:a0c63ad9-ce02-4cc8-9264-c66d35ec8fa1`.

[93] W. Tao, P. Zhu, C. Xu, and Z. Liu. Uncertainty quantification of mechanical properties for three-dimensional orthogonal woven composites. part ii: Multiscale simulation. *Composite Structures*, 235:111764, 2020. doi: 10.1016/j.compstruct.2019.111764.

[94] D. Trias, J. Costa, A. Turon, and J. E. Hurtado. Determination of the critical size of a statistical representative volume element (srve) for carbon reinforced polymers. *Acta Materialia*, 54(13):3471–3484, 2006. doi: 10.1016/j.actamat.2006.03.042.

[95] 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. doi: 10.1016/j.mechmat.2005.10.003.

[96] S. Turteltaub and G. de Jong. Multiscale modeling of the effect of sub-ply voids on the failure of composite materials. *International Journal of Solids and Structures*, 165:63–74, 2019. doi: 10.1016/j.ijsolstr.2019.01.031.

[97] S. Turteltaub and R. Suárez-Millán. Energetically-consistent multiscale analysis of fracture in composites materials. *European Journal of Mechanics - A/Solids*, 84:104079, 2020. doi: 10.1016/j.euromechsol.2020.104079.

[98] S. Turteltaub, N. van Hoorn, W. Westbroek, and C. Hirsch. Multiscale analysis of mixed-mode fracture and effective traction-separation relations for composite materials. *Journal of the Mechanics and Physics of Solids*, 117:88–109, 2018. doi: 10.1016/j.jmps.2018.04.009.

[99] S. Y. Varnamkhasti. *Multiscale thermomechanical analysis of multiphase materials*. PhD thesis, Delft University of Technology, 2015.

[100] T .J. Vaughan and C. T. McCarthy. A combined experimental–numerical approach for generating statistically equivalent fibre distributions for high strength laminated composite materials. *Composites Science and Technology*, 70(2):291–297, 2010. doi: 10.1016/j.compscitech.2009.10.020.

[101] T. J. Vaughan and C. T. McCarthy. A micromechanical study on the effect of intra-ply properties on transverse shear fracture in fibre reinforced composites. *Composites Part A: Applied Science and Manufacturing*, 42(9):1217–1228, 2011. doi: 10.1016/j.compositesa.2011.05.004.

[102] L. Wan, Y. Ismail, C. Zhu, P. Zhu, Y. Sheng, J. Liu, and D. Yang. Computational micromechanics-based prediction of the failure of unidirectional composite lamina subjected to transverse and in-plane shear stress states. *Journal of Composite Materials*, 54(24):3637–3654, 2020. doi: 10.1177/0021998320918015.

[103] Z. Wang, X. Wang, J. Zhang, W. Liang, and L. Zhou. Automatic generation of random distribution of fibers in long-fiber-reinforced composites and mesomechanical simulation. *Materials & Design*, 32(2): 885–891, 2011. doi: 10.1016/j.matdes.2010.07.002.

[104] K. Woo and D. S. Cairns. Selective activation of intrinsic cohesive elements for fracture analysis of laminated composites. *Composite Structures*, 210:310–320, 2019. doi: 10.1016/j.compstruct.2018.11.055.

[105] M. Xu. *Concurrent coupling of atomistic and continuum models*, chapter 4, pages 93–133. Oxford University Press, 2009. ISBN 9780199233854.

[106] L. Yang, Y. Yan, Y. Liu, and Z. Ran. Microscopic failure mechanisms of fiber-reinforced polymer composites under transverse tension and compression. *Composites Science and Technology*, 72(15):1818–1825, 2012. doi: 10.1016/j.compscitech.2012.08.001.

[107] L. Yang, Y. Yan, Z. Ran, and Y. Liu. A new method for generating random fibre distributions for fibre reinforced composites. *Composites Science and Technology*, 76:14–20, 2013. doi: 10.1016/j.compscitech.2012.12.001.

[108] L. Yang, Z. Wu, Y. Cao, and Y. Yan. Micromechanical modelling and simulation of unidirectional fibre-reinforced composite under shear loading. *Journal of Reinforced Plastics and Composites*, 34(1):72–83, 2015. doi: 10.1177/0731684414562873.

[109] L. Zhao, Y. Gong, J. Zhang, Y. Chen, and B. Fei. Simulation of delamination growth in multidirectional laminates under mode i and mixed mode i/ii loadings using cohesive elements. *Composite Structures*, 116:509–522, 2014. doi: 10.1016/j.compstruct.2014.05.042.

A

Database parameters

Table A.1: Database Parameters.

ENTRY	DESCRIPTION
ref	Test reference number
status	Current status in terms of progress
description	Test name/description
id_group	ID of batch/type of analysis being performed
grid_size	Mesh size parameter for Gmsh
reps	Statistical repetition number for that test
load_angle	Angle of rotation of strain tensor
viscreg	Viscosity used to stabilize cohesive elements
lastframe	Time of last converged frame in analysis
geo_path	Path to .geo file
mesh_path	Path for .msh file of mesh before fixing procedure
mesh_fixed_json	Path to json with fixed mesh data
RVE_ply_orientations	Ply angles list. First ply is the bottom ply
RVE_w	Width of RVE
RVE_l_default	Default value for RVE length in case only 0° or 90° fibers are present
RVE_l	Length of the RVE used in the final geometry
RVE_t_plies	List with thickness of each ply
RVE_t	Total thickness of RVE
RVE_V_f	Fiber volume fraction target of RVE
RVE_f_diameter	Fiber diameter
RVE_V_f_after_geo	Actual fiber volume fraction on each ply after geometry generation
RVE_bc	Boundary conditions of RVE for geometry
INP_bc	Boundary conditions of RVE for analysis
e11,e12,e13,e21,e22,e23,e31,e32,e33	Macroscopic strain tensor components
rho_f	Fiber density
E1_f, E2_f, E3_f	Fiber's elastic modulus values
nu12_f, nu13_f, nu23_f	Fiber's Poisson's coefficients
G12_f, G13_f, G23_f	Fiber's shear modulus values
rho_m	Matrix density
nu_m	Matrix Poisson's ratio
E_m	Matrix elastic modulus
dam_init_cf	Damage initiation criteria for fibers
dam_evol_cf	Damage evolution criteria for fibers
t_ult_cf	Ultimate traction of fibers
G_f_cf	Fracture toughness of fibers

Continuation of Table A.1	
ENTRY	**DESCRIPTION**
d_f_cf	Fiber failure opening
E_coh_cf	Cohesive stiffness of fibers
dam_init_cm	Damage initiation criteria for matrix
dam_evol_cm	Damage evolution criteria for matrix
t_ult_cm	Ultimate traction of matrix
G_f_cm	Fracture toughness of matrix
d_f_cm	Matrix failure opening
E_coh_cm	Cohesive stiffness of matrix
dam_init_ci	Damage initiation criteria for interface
dam_evol_ci	Damage evolution criteria for interface
t_ult_ci	Ultimate traction of interface
G_f_ci	Fracture toughness of interface
d_f_ci	Interface failure opening
E_coh_ci	Cohesive stiffness of interface

NOTE: Some of the parameters are legacy from the previous 2D implementations. These were kept because the scripts which use them have not yet been completely changed and properly tested.

B

Test parameters

The parameters that are common for all tests are shown in Table B.1. The remaining parameters are presented in Table B.2.

μ	**RVE w**[mm]	**RVE l**[mm]	**RVE t**[mm]	**INP BC**
0.0001	0.04	0.04	0.04	[P,P,P,P,P,P]

Table B.1: Test parameters that are common for all tests.

					Applied Strain Tensor								
ID	**Layup**	V_f	**Fiber Diameter**[μm]	**RVE BC**	11	12	13	21	22	23	31	32	33
1	[0,90,0]	0.30	5.5	[W,W,W,W,W,W]	0	0	0	0	0.3	0	0	0	0
3	[90]	0.15	7.0	[P,P,P,P,P,P]	0	0	0	0	0.3	0	0	0	0
6	[0,90,0]	0.30	5.5	[W,W,W,W,W,W]	0.3	0	0	0	0	0	0	0	0
8	[0]	0.30	5.5	[W,W,W,W,W,W]	0.3	0	0	0	0	0	0	0	0
9	[0]	0.30	5.5	[W,W,W,W,W,W]	0	0	0	0	0	0	0	0	0.3
10	[0]	0.30	5.5	[W,W,W,W,W,W]	0.3	0	0	0	0.3	0	0	0	0
11	[0,90,0]	0.30	5.5	[W,W,W,W,W,W]	0	0	0	0	0.3	0	0	0	0.3
12	[0]	0.30	5.5	[W,W,W,W,W,W]	0	0	0	0	0.3	0	0	0	0
13	[0]	0.30	5.5	[W,W,W,W,W,W]	0	0	0	0	0.3	0	0	0	0
14	[0,90,0]	0.30	5.5	[W,W,W,W,W,W]	0	0	0	0	0.3	0	0	0	0

Table B.2: Remaining test parameters.