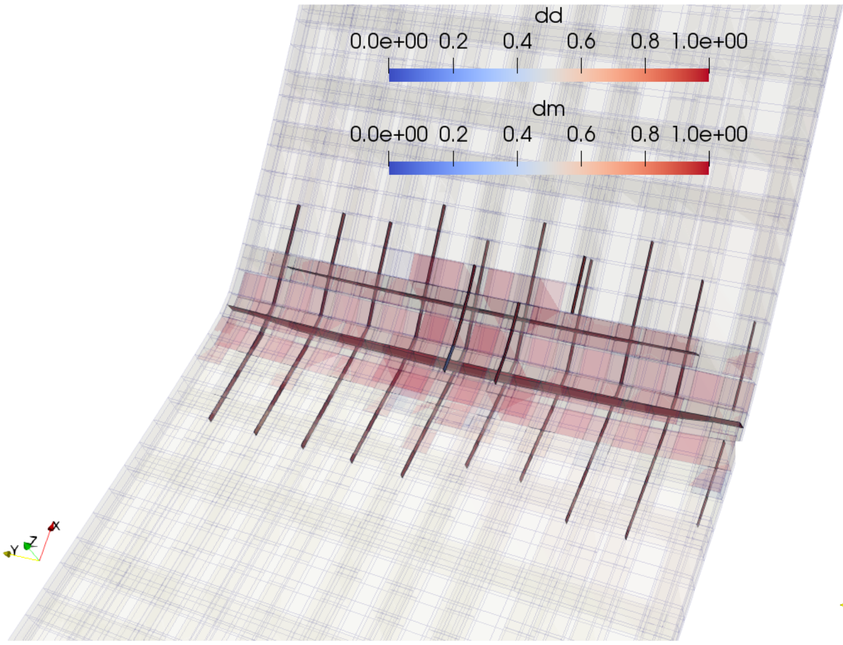
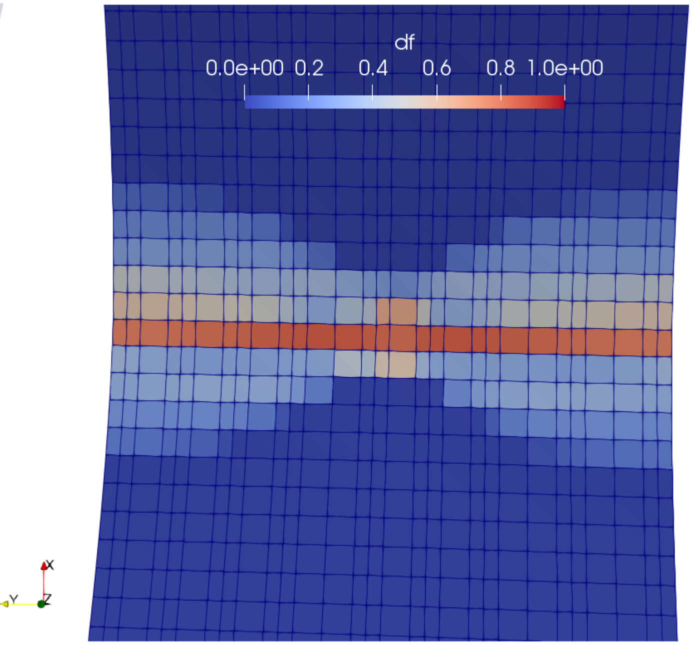
A co-rotational floating node finite element for geometric non-linear fracture modelling of composite structures

Master of Science Thesis

O. Zahalka

Technische Universiteit Delft





Delft University of Technology

Challenge the future

A co-rotational floating node finite element for geometric non-linear fracture modelling of composite structures

Master of Science Thesis

by

O. Zahalka

in partial fulfillment of the requirements for the degree of

Master of Science
in Aerospace Structures and Materials

at the Delft University of Technology,
to be defended publicly on Monday September 17, 2018 at 14:00.

Supervisor:	Dr. B. Y. Chen,	Aerospace Structures and Materials
Thesis committee:	Dr. S. R. Turteltaub,	Aerospace Structures and Materials
	Dr. S. J. Hulshoff,	Aerodynamics

TU Delft Delft University of Technology

Abstract

Fibre-reinforced composites have become increasingly attractive for many engineering applications in the last decades. A very interesting aspect of these materials is that their mechanical properties can be tailored for optimum strength and stiffness by controlling the orientation of the fibers embedded in the matrix material. Composites are characterized by high strength properties, strong corrosion resistance, improved damage tolerance and can lead to considerable weight and cost reduction when compared to their metallic counterparts. However, accurate modelling of damage in composites is still an active research topic, as their progressive failure involves the interaction of various intra- and inter-laminar damage mechanisms, which often lead to complex fracture paths. To this regard, the Floating Node Method (FNM) proved to be particularly suited for the modelling of complex cracking scenarios within a finite element. This thesis work investigates the modelling of geometric non-linear three dimensional fracture problems in composites using the Floating Node Method. A co-rotational approach is proposed as a convenient, conceptually simple way to include geometric non-linear effects in problems characterized by large rotations but small deformations. This approach allows re-use of the conventional linear FEM formulation by separating rigid body and purely deformational motions at the element level. The co-rotational procedure, first implemented and validated on a linear solid brick element, is subsequently applied to the Floating Node (FN) element. Finally, a demonstration of the element's potential in capturing geometric non-linear effects is offered, addressing the modelling of crush and impact loading on composite laminates.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms

CFRP	Carbon Fiber Reinforced Polymers
CR	Co-Rotational
CZM	Cohesive Zone Modelling
DCM	Discrete Crack Models
DOF	Degree Of Freedom
EICR	Element Independent Co-Rotational procedure
FE	Finite Element
FEM	Finite Element Method
FN	Floating Node
FNM	Floating Node Method
FRP	Fiber Reinforced Polymers
HHT	Hilber-Hughes-Taylor
LEFM	Linear Elastic Fracture Mechanics
LHS	Left Hand Side
PK1	First Piola-Kirchhoff
PK2	Second Piola-Kirchhoff
PN	Phantom Node
PNM	Phantom Node Method
RHS	Right Hand Side
RP	Reference Point
SCM	Smeared Crack Models
SE	Sub-Element
SVD	Singular Value Decomposition
TL	Total Lagrangian
UEL	User ELement
UL	Updated Lagrangian
VCCT	Virtual Crack Closure Technique
VTK	Visual ToolKit
XFEM	eXtended Finite Element Method

Abaqus Elements

C3D8	3D 8-node solid full integration element
C3D8I	3D incompatible mode 8-node solid element
C3D8R	3D 8-node solid reduced integration element
CPS4	2D 4-node plane stress full integration element
S4	2D 4-node shell full integration element
S4R	2D 4-node shell reduced integration element

Greek Symbols

α,β,γ	control parameters for dynamic procedures
δ	separation in the cohesive softening law
δ_{eff}	effective separation
Γ	boundaries of a general domain
Γ_{t}	boundaries of a domain on which the traction is prescribed
Γ_{u}	boundaries of a domain on which the displacement is prescribed
λ_1,λ_2,λ_3	eigenvalues obtained through spectral decomposition of a 3×3 matrix
μ_{d}	damping coefficient
μ_{v}	viscous parameter for the viscous regularization
ν	Poisson ratio
Ω	physical domain in the current configuration
Ω_0	physical domain in the reference configuration
Ω_α	partition of the finite element domain in the FNM, with $\alpha = \mathrm{A}, \mathrm{B}$
Ω_{e}	physical domain of a finite element
Φ	mapping function
$\boldsymbol{\epsilon}$	strain tensor
$\boldsymbol{\Lambda}$	diagonal matrix formed with singular values in eigenvalues in the spectral decomposition
$\boldsymbol{\Omega}_\ell$	local spin matrix
$\boldsymbol{\Sigma}$	diagonal matrix formed with singular values in SVD
$\boldsymbol{\sigma}$	Cauchy stress tensor
$\boldsymbol{\tau}$	traction vector in a cohesive element
$\boldsymbol{\theta}$	pseudo-vector
ρ	density
τ	traction in the cohesive softening law
Θ	absolute temperature
$\varepsilon_{\mathrm{ij}}$	strain component ij, with $\mathrm{i}, \mathrm{j} = 1, 2, 3$

ξ, η, ζ	isoparametric natural coordinates of an element

Operators

δ_{ij}	Kronecker delta function
$\nabla(\bullet)$	gradient operator
$\nabla \cdot (\bullet)$	divergence operator
$\mathbf{A} : \mathbf{B}$	contraction or inner product between tensor $\mathbf{A}$ and $\mathbf{B}$
$\mathbf{a} \cdot \mathbf{b}$	internal product between vector $\mathbf{a}$ and $\mathbf{b}$
$\mathbf{a} \times \mathbf{b}$	external product between vector $\mathbf{a}$ and $\mathbf{b}$
$\mathrm{asym}(\bullet)$	function that returns the anti-symmetric part of a matrix
$\mathrm{axial}(\bullet)$	function that extracts the pseudo-vector from a spinor argument
$\det(\bullet)$	function that calculates the determinant of a matrix
$\mathrm{e}^{(\bullet)}$	matrix exponential function with $(\bullet)$ being a square matrix
$\log_{\mathrm{e}}(\bullet)$	matrix natural logarithm function
$\mathrm{spin}(\bullet)$	function that construct a spinor from an axial vector argument
$\mathrm{sym}(\bullet)$	function that returns the symmetric part of a matrix

Roman Symbols

$\ddot{\mathbf{u}}$	acceleration vector
$\Delta\mathrm{t}$	time increment
$\Delta\mathrm{t}_{\mathrm{crit}}$	critical time increment
$\delta\mathrm{W}$	virtual work
$\delta\mathbf{u}$	displacement variation vector
$\delta\mathbf{v}$	virtual displacements
$\dot{\mathbf{u}}$	velocity vector
ℓ	length
ℓ_{e}	characteristic element length
ℓ_{e}	element characteristic length
f	general function
$\mathbf{\Delta R}$	residual load vector
$\mathbf{B}$	finite element strain-displacement matrix
$\mathbf{b}$	body forces
$\mathbf{C}$	damping matrix
$\mathbf{D}$	rate of deformation tensor/material constitutive tensor
$\mathbf{E}$	Green strain tensor
$\mathbf{e}$	base vector of a reference system

$\mathbf{E}^*$	Almansi strain tensor
$\mathbf{e}_\mathrm{n}$,$\mathbf{e}_\mathrm{t}$,$\mathbf{e}_\ell$	normal, tangent transverse and tangent longitudinal vectors to the mid-plane surface of a cohesive element
$\mathbf{F}$	deformation gradient
$\mathbf{f}$	general force vector
$\mathbf{f}^\mathrm{ext}$	external force vector
$\mathbf{f}^\mathrm{int}$	internal force vector
$\mathbf{f}^\mathrm{kin}$	kinetic force vector
$\mathbf{I}$	identity matrix
$\mathbf{J}$	finite element Jacobian matrix
$\mathbf{K}$	finite element stiffness matrix
$\mathbf{K}_\mathrm{tg}$	geometric contribution to the element tangent stiffness
$\mathbf{L}$	velocity gradient
$\mathbf{M}$	finite element mass matrix
$\mathbf{N}$	finite element shape function matrix
$\mathbf{n}$	normal vector
$\mathbf{P}$	First Piola-Kirchhoff stress tensor/orthogonal matrix obtained with SVD
$\mathbf{p}$	linear momentum
$\mathbf{Q}$	general orthogonal tensor
$\mathbf{q}$	heat flux vector
$\mathbf{R}$	transformation rotation matrix
$\mathbf{S}$	Second Piola-Kirchhoff stress tensor/generic skew symmetric matrix
$\mathbf{T}$	transformation matrix in the CR formulation
$\mathbf{t}$	traction vector/surface forces
$\mathbf{U}$	right stretch tensor
$\mathbf{u}$	displacement vector
$\mathbf{V}$	left stretch tensor
$\mathbf{v}_1$,$\mathbf{v}_2$,$\mathbf{v}_3$	eigenvectors obtained through spectral decomposition of a 3×3 matrix
$\mathbf{W}$	spin tensor/orthogonal matrix obtained with SVD
$\mathbf{X}$	material coordinates of a point
$\mathbf{x}$	current position vector of a point
$\mathbf{x}_\mathrm{mid}$	mid-plane nodal coordinates of a cohesive element
A	area
$\mathrm{array}_\mathrm{DOF}$	DOF array for a generic finite element
b	width

c	generic crack node generated by element partition in the FNM
d	generic damage parameter
E	Young’s modulus
$\mathrm{e_i}$	i-th edge of a finite element
$\mathrm{f_f}$	failure index for the fibre damage
$\mathrm{f_m}$	failure index for the matrix damage
$\mathrm{G_c}$	fracture toughness
$\mathrm{G_n}$,$\mathrm{G_t}$,G_ℓ	normal tensile, transverse shear and longitudinal shear strain energy density
$\mathrm{G^1_{fc,T}}$,$\mathrm{G^1_{fc,C}}$	single ply tensile/compressive fibre fracture toughness
$\mathrm{G_{ij}}$	shear modulus in the plane ij, with $\mathrm{ij} = 12, 23, 13$
h	height
J	Jacobian or determinant of the deformation gradient
K	penalty stiffness of a cohesive element
m	mass
N	finite element shape function
$\mathrm{n_N}$	number of nodes in a finite element
$\mathrm{n_{const}}$	number of equations deriving from constitutive relation
$\mathrm{n_{dim}}$	number of dimensions
$\mathrm{N_{DOF}}$	number of DOF
$\mathrm{n_e}$	number of elements
$\mathrm{q_s}$	power supplied by heat sources per unit mass
s	specific entropy
$\mathrm{S_t}$,S_ℓ	transverse and longitudinal shear strength
t	time
$\mathrm{t_{ply}}$	single ply thickness
$\mathrm{w_{int}}$	internal energy per unit mass
$\mathrm{X_t}$,$\mathrm{X_c}$	longitudinal tensile/compressive strength, respectively
$\mathrm{Y_t}$,$\mathrm{Y_c}$	transverse tensile/compressive strength, respectively

Sub/superscripts

$\bullet^{\mathrm{d}}$	related to the damaged elastic constants values
$\bullet^{\mathrm{top}}$,$\bullet^{\mathrm{bot}}$	related to the top/bottom SEs of a cohesive FN element
$\bullet^{\mathrm{v}}$	related to regularized values in the viscous regularization of damage
$\bullet_1$,$\bullet_2$,$\bullet_3$	related to global x, y, z directions, respectively/related to fiber, in plane transverse and out of plane transverse directions of the composite, respectively
$\bullet_\ell$	related to local quantities

$\bullet_\Omega$	related to finite element with domain Ω
$\bullet_\xi, \bullet_\eta, \bullet_\zeta$	related to derivatives with respect to the natural coordinates
$\bullet_\mathrm{g}$	related to global quantities
$\bullet_\mathrm{n}$	related to normal direction
$\bullet_\mathrm{o}$	related to reference configuration
$\bullet_\mathrm{Q}$	related to integration points
$\bullet_{\Gamma_\mathrm{c}}$	related to cohesive element with boundaries Γ_c in the FNM
$\bullet_\mathrm{CE}$	related to cohesive elements
$\bullet_\mathrm{coh}$	related to cohesive elements
$\bullet_\mathrm{n}, \bullet_\mathrm{t}, \bullet_\mathrm{s}$	related to the normal, transverse shear and longitudinal shear components, respectively
$\bullet_\mathrm{ply}$	related to a ply in the FN laminate element
$\bullet_\mathrm{SE}$	related to SE in the FNM
$\hat{\bullet}$	related to co-rotational quantities

1

Introduction

Composites are very attractive materials, mainly due to their excellent strength-to-weight ratios, improved damage tolerance, corrosion resistance and design flexibility. In the very beginning, their application has been confined to secondary structures, as there was a poor understanding of the failure mechanisms involved. However, in the last three decades, the use of composites has greatly increased in many engineering fields, especially in aircraft structures, thanks to the advances in manufacturing processes. As our understanding in this field is improving, there is still the need for novel, less computationally expensive and more accurate methods to address non-linear fracture modelling in composite structures in order to increase reliability and reduce the cost given by destructive testing.

Efficient modelling of non-linear effects in FE formulations is a topic that has been addressed by many researchers, but still deserves further developments. Even though the linear theory provides satisfying results for most real-world applications, every structure exhibits a certain degree of non-linearity. As reported by Belytschko et al., four different sources of non-linearity exist: material non-linearity, geometric non-linearity and non-linearities arising from either force or displacement boundary conditions [2]. Considering the modelling of geometric non-linear effects, those characterized by small strains but large displacements/rotations are particularly relevant to composite structures, as these materials are known to experience small strains up to failure. Existing FE software packages already offer the possibility to account for geometric non-linear effects, even though the formulations currently embedded in the software are expensive from a computational point of view [3].

In a similar fashion, modelling of fracture initiation and propagation is an active ongoing research topic, especially in composite structures. Damage evolution in composites is far more complicated than in other materials, since the former ones are characterized by several failure mechanisms that usually occur simultaneously [4]. Among these, the most important ones are fibre breakage, delamination, which refers to debonding of adjacent layers, and matrix cracks, the latter being the most common failure mechanism [5]. The main challenge in FEM modelling consists in developing an approach that accounts for mutual interaction between different failure mechanisms.

1.1. State of the art

1.1.1. Modelling of geometric non-linearities

In order to include geometric non-linear effects, three different approaches have been found to be used in the literature: Total Lagrangian (TL), Updated Lagrangian (UL) and Co-rotational (CR).

In the TL approach , all the kinematic variables are referred to the initial or undeformed configuration. Alternatively, these ones can be referred to the last converged configuration, which is assumed as the new reference configuration in the UL approach. The latter was found to be preferred over the former to account for geometric non-linear effects [6–13]. The main reason is that the strain-displacement matrix, which describes the kinematic relations in the structure, has a more complicated form in the TL approach when compared to the UL approach, due to the presence of the term including the initial strain effect [9].

However, the relatively new CR approach has been found to be particularly convenient for the modelling of geometric non-linearities. The basic idea behind this approach is to define a local reference system for every element of the discretized structure, which is given the name of *co-rotational frame*. This reference configuration translates and rotates with the element, without deforming, so that orthogonality is maintained [14]. In this formulation, a decomposition of the total displacement is performed at the element level. The total displacement is decomposed into rigid body motion and deformational components using geometric methods. The mathematics involved in this procedure is described by Argyris in his compendium on finite rotations [15]. The afore mentioned orthogonality of the frame ensures exact decomposition of rigid body and deformational motions [3].

If deformational displacements and rotations are small, relative to the CR frame, the standard linear formulation can be used. The geometric non-linearity is subsequently accounted for via the transformation of the results from the CR to the global frame [16]. In order to use the linear FE formulation, deformational displacements and rotations must remain small relative to the element dimensions, therefore the CR frame is updated for each load increment [17]. The main advantage of this approach is that only the conventional linear strain-displacement matrix is used and no further consideration has to be made on the element [18].

This approach has been widely implemented by many researchers for beam and shell elements [17, 19, 20]. Crisfield developed the concept of *consistent linearization* of the stiffness matrix, as a true variation of the internal force vector, and extended the formulation to three dimensional solid elements [16, 18, 21, 22]. This procedure was found to be particularly suitable in geometric non-linear problems characterized by large rotations but small strain [3, 16, 17, 23–25].

A noteworthy contribution to the development of the co-rotational theory was brought by Rankin and Brogan, who are considered to be the pioneers of the so called *Element Independent Co-Rotational procedure* (EICR). These researchers developed a CR procedure that allows re-use of existing linear finite elements [24]. As observed by Felippa et al. [3], the EICR formulation is particularly interesting because it can be easily embedded in existing FEM programs. The EICR can be thought of as a *front end filter*, which is lying in between the assembler/solver and the existing FE library. The filtering operation is purely geometric and works by addition and removal of rigid body motion, as it is shown in Figure 1.1.

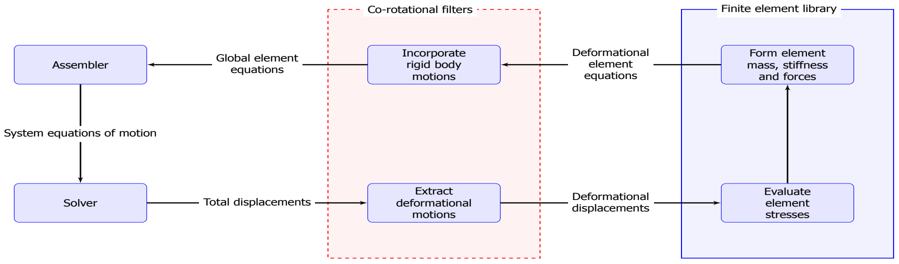


Figure 1.1: Flowchart illustrating the EICR formulation

The CR procedure proved to have several advantages over the TL/UL approaches [3, 16, 26]. Due to the fact that linear FE assumptions are valid in the CR frame, the resulting formulation is much simpler. At the same time, when using co-rotated quantities, which are frame invariant, objectivity is satisfied, so that the material constitutive relations remain unchanged due to a change of observer [2]. This method is also directly providing the stress components oriented with the material, which are those of main interest. As a result, the post-processing is simplified with this formulation [2]. Finally, the most important advantage of this formulation, is its flexibility of use. Since it is an *element independent* procedure, this technique offers a relatively simple way to embed geometric non-linear effects in the ”most complex” linear finite elements.

1.1.2. Modelling of fracture

Numerical modelling of fracture experienced a fast advancement in the last decades, becoming more and more reliable thanks to the development of several fracture modelling techniques. In a first instance, the so called *Discrete Crack Models* were used (DCMs). In such approach, the crack is modelled geometrically, in the idealization of the structure. On the other hand, in *Smeared Crack Models* (SCMs), the damage is incorporated by deteriorating the local stiffness of the element when integration is carried out at the integration point. A downside of the DCM is that crack initiation cannot be predicted, as the crack is idealized as a geometric defect at the beginning of the analysis. Furthermore, a *mesh bias* is introduced, because the crack is forced to grow along the element boundaries. Remeshing techniques help to overcome the numerical difficulty arising from a continuous change in topology [27]. However, a draw back of remeshing in this case is given by the constant change of mesh connectivity. Although these methods offer reliable results, they were found to have some difficulties in capturing the interaction between different cracks, other than having a considerably high mesh dependency. Therefore, they are inherently affected by limitations when it comes to model damage in composites, where capturing the interaction between matrix cracks and delamination is particularly important [28].

Within the standard FEM framework, the *Cohesive Zone Modelling* approach (CZM) is a widely used technique. In this approach, the non-linear fracture behavior is simplified specifying traction-separation laws along fracture surfaces, through the use of purposely designed interface elements called *cohesive elements*. The main advantage of this technique over the previous ones is that it may be used to model crack nucleation. Regarding the crack propagation direction, this is still limited to follow the boundaries of the elements. Therefore, in order to have satisfying results, the crack path should be known in advance, as cohesive elements have to be inserted a priori in the FE mesh [29, 30]. Concerning composite laminates, this technique is widely used to capture the delamination damage, as it is known that this is confined to the interfaces between the plies [29, 31].

All the previously introduced techniques are characterized by a severe drawback: mesh dependency. The crack is forced to grow along the boundaries of the elements, so that the crack path cannot be determined with a particularly high accuracy. An important tipping point in fracture modelling has been reached through the development of the so called *enriched methods*. This class of modelling techniques allow an arbitrary crack growth, potentially inside the elements, through the introduction of additional nodes and degrees of freedom (DOFs) in the finite element. To this regard, three methods are worth discussing: the eXtended Finite Element Method (XFEM), the Phantom Node Method (PNM) and the Floating Node Method (FNM).

XFEM is a method based on the partition of unity concept and has been developed from the work of Belytschko and Black [32] and Moës and Belytschko [33]. The key idea behind XFEM is to introduce additional degrees of freedom for the nodes and to use special *enrichment functions* to better approximate the displacement field given the presence of discontinuities in the structure. The enrichment functions choice is made on the a priori solution to the problem. In the case of crack analysis, the enrichment is aimed at higher accuracy in the numerical approximate solution, using information extracted from the analytical solution in Linear Elastic Fracture Mechanics (LEFM) theory. In XFEM, the enrichment is activated only at the nodes of the elements that are intersected by the crack or in its neighbourhood, so that the standard FEM formulation is applied to the rest of the model. This method may be used to model both fracture initiation and propagation without knowing a priori the crack path. The main advantage is that, contrary to previously used fracture modelling techniques, the need for re-meshing is eliminated and a discontinuity can propagate inside a finite element [32, 33]. A drawback of this method is that standard Gauss integration cannot be used on the cracked sub-elements (SEs), although purposely designed partial integration techniques are available. Moreover, an intrinsic error exists in the method, because the crack geometry is approximated as straight when mapped from the physical to the natural space of the element, even though it is generally not the case [34]. At the same time, this method proved to have some difficulties in capturing the mutual interaction when dealing with multiple cracks [35]. Since its early development, this technique has been used to model damage in composite structures by many authors in literature. It proved to be able to capture the interaction between matrix crack and interlaminar damage [36–38] and to be suitable for the modelling of damage in micro-scale models [39, 40] as well as in multi-scale models [41].

The Phantom Node Method (PNM) has been proposed for the first time by Hansbo and Hansbo [42]. In this method the potential discontinuity is incorporated through insertion of additional *phantom nodes*, located on top of the real nodes of the element. If a certain criterion is met, the discontinuity is created and the element splits into two SEs. In such circumstances, the element displacement field is replaced by two displacement fields related to the two newly defined SEs. Similarly to XFEM, numerical integration is performed on the partially active sub-domains of the initial element. The PNM has proven to be equivalent to the XFEM with only the Heaviside enrichment function, when modelling a strong discontinuity [43]. On the other hand, the differences between the PNM and XFEM are [28, 34]:

- In the PNM the number of nodes is doubled, as two SEs are formed, while in XFEM the number of nodes remains constant.
- The PNM is using only standard shape functions for the displacement field interpolation, while XFEM is using extra enrichment functions.
- The PNM is relating only displacement DOFs to each node, while XFEM is introducing additional DOFs at every node that do not have a proper physical meaning.
- The PNM is easier to implement in standard FEM packages when compared to XEFM due to the fact that only standard shape functions for the displacement approximation.

The PNM is proven to have good accuracy in fracture modelling problems. Concerning damage modelling in composite structures, this method has been applied to cohesive elements to better capture delamination damage [44–47], but it has also been used to model the interaction between delamination and matrix cracks [28, 48, 49]. Although really satisfying results were obtained, the PNM is still affected by the same error on the crack geometry when this one is mapped from the physical to the natural space [34]. Similarly to XFEM, while the PNM proved to be suited to model a single discontinuity in a finite element, it showed difficulties when dealing with multiple discontinuities [34].

The Floating Node Method (FNM) is a relatively new enriched method that was introduced by Chen et al. [34]. This method shares some similarities with the previously introduced PNM. A finite element in the FNM formulation is characterized by the standard FEM real nodes plus a convenient number of *floating nodes*. The latter ones are given this name because they do not need an associated geometric location before the discontinuity is predicted. The exact number of floating nodes to be used depends on the number and type of discontinuities desired to be modelled [50]. This method arose from the idea that, when an element is cut from a discontinuity, the nodal position associated with the phantom DOFs is not the most suitable in terms of transformation to the natural coordinates, nor in terms of integration [34].

Given a certain criterion for the discontinuity, if this one is not met, the FNM will function just as the standard FEM formulation. In this case, the floating DOF sets are not used [51]. On the contrary, when the criterion is satisfied, the element is partitioned into two or more SEs, depending on the nature of the discontinuity. In such case, the floating nodes assume a specific location along the element edges cut by the crack, in order to form the boundaries of the discontinuity and the domains of the SEs. At the same time, the floating DOFs are directly used to represent the displacements at the crack nodes. Since the partition domains are fully defined, thanks to the active role played by the floating nodes, no partial integration is required on the element domain [34].

Cohesive cracks can also be modeled with the FNM. As a matter of fact, when the FN element is partitioned into SEs, a cohesive element is inserted in between the SEs. This feature makes the FNM very suitable for damage modelling in composites, since delamination and matrix cracks are explicitly represented with cohesive elements. Furthermore, through the introduction of the novel *edge status variable* in the element formulation, all the elements can share information regarding the location of the discontinuity, allowing cohesive cracks to propagate and partitioning a specific element accordingly [51]. A more accurate and simpler representation of the discontinuities is obtained with the FNM, which is able to capture more complex cracking scenarios, such as T-cracks or an intersecting network of cracks. At the same time, if a weak discontinuity is to be modelled, the FNM requires less DOFs compared to the PNM [34].

Most importantly, the equilibrium equations that are obtained through the FNM are exactly the ones that would be obtained with an analogous FE mesh in which the discontinuities are included in the idealization of the model. As a result, standard Gauss integration can be used, leading to improved accuracy. Finally, as it is made direct use of the crack coordinates, the error in the approximation of the discontinuity when mapping from physical to natural coordinates is mitigated compared to XFEM and PNM [34].

1.2. Motivation and scope of the thesis

In conclusion to the literature research, it is believed that the development of an approach combining the CR formulation with the FNM would be promising to address geometric non-linear fracture problems in composites, such as low-velocity impacts and crushes.

The main reason for choosing the CR approach is that it is an element independent procedure. Once the formulation is implemented for a finite element, it can be easily extended to elements belonging to the same class. This feature is particularly convenient, as the FN element can split into different SEs. Another advantage of this approach is that it uses the conventional linear FEM formulation, resulting in an overall simplified implementation and a less complicated form of the element strain-displacement matrix. Furthermore, the CR approach was found to be mainly used in problems characterized by small strains but large rotations [3]. For this reason it is believed to be very suitable for composite structures, which do not undergo large deformations up to failure, but may experience large rotations. This method is also preferred over the TL/UL approaches because it is using frame invariant strain/stress quantities, so that objectivity is satisfied [2]. Finally, post-processing is simplified, as this approach is directly providing the stress measures of main interest in the structure.

Comparing different fracture modelling techniques, the FNM proved to have considerable advantages over the other enriched methods, especially when it comes to model fracture in composites. The innovative use of edge status variables together with the integration of cohesive SEs in the formulation allow the FN element to capture many different cracking scenarios and to better represent the mutual interaction between matrix cracking and delamination [34].

As a result, the main research objective that has been set for this project is the following:

To implement a floating node element with geometric non-linear capabilities via a co-rotational approach

Such element should be able to provide high-fidelity simulations of damage initiation and evolution in composite laminates in geometric non-linear problems. To this regard, the following sub-objectives have been set:

- To implement and validate a linear solid element.
- To embed such an element in a CR formulation and validate its geometric non-linear capabilities.
- To integrate the FN element in the CR procedure to model geometric non-linear fracture in composites.

1.3. Structure of the report

The content of the report has been divided into five different chapters, each containing several sections.

Chapter 2 is meant to provide the reader basic knowledge of continuum mechanics and Finite Element Method theory. Starting with the description of motion, most used strain/stress measures and conservation laws are presented, to end with a note on the concepts of material frame indifference and frame invariance. From the strong form of the governing equations, a weak form is derived, leading to the discretized form of the governing equations used in the Finite Element Method. The concept of geometric non-linearity is introduced to the reader, reproducing a popular geometric non-linear benchmark to give more insight into the modelling of such problems in a FEM framework. At the end of the chapter, the detailed implementation of the linear 8-node hexahedral element for static procedures is illustrated and validated.

Following the implementation of the linear element, the CR procedure is reported in Chapter 3. In this chapter, relevant theoretical background on finite rotations is first given to the reader. Next, the CR implementation is presented in details and validated with popular geometric non-linear benchmarks.

Chapter 4 starts with an overview of the FNM, extensively describing its partitioning algorithm and implemented damage initiation/propagation criteria. At the end of the chapter, the combination of the FNM and the CR procedure is proposed, offering a demonstration of the element's potential in addressing geometric non-linear problems such as crushes and impacts.

Finally, conclusions and recommendations are given in Chapter 5. At the end of the report, annexes that are meant to give support to the main report can be found in Appendix A.

2

Theoretical background

Before digging into the content of the thesis, it is crucial to introduce the relevant theoretical background. First, fundamental concepts of continuum mechanics theory, needed to kinematically characterize a continuous body, are given in Section 2.1. These are used to define the conservation laws governing the motion of continua. At the end of the section, the concept of material frame indifference and frame invariance are introduced to the reader. Secondly, the FEM formulation to obtain approximate solutions to the field governing equations is outlined in Section 2.2. A popular geometric non-linear benchmark problem is proposed at the end of the section, to highlight relevant features regarding FEM modelling of such problems. Lastly, the 8-node linear hexahedral finite element is presented in Section 2.3. This section gives the topological definition of the element, presents its FE formulation and provides an insight on the implementation of a FE using the User ELement (UEL) Abaqus subroutine. At the end of the section, the implemented user element is validated in comparison to the C3D8 Abaqus element.

2.1. Continuum mechanics fundamentals

Continuum mechanics is a specific branch of mechanics that studies the behavior of continuous media. The basic assumption of this theory is to consider a body as completely filled by matter, therefore neglecting its atomic structure. In this section, relevant continuum mechanics notions are provided to the reader, omitting their detailed derivations, which can be found in most famous academic books on this matter [2, 52–54]. The section begins with a description of motion and deformation, with an emphasis on the difference between the Lagrangian and Eulerian description. Next, the most common strain and stress measures are presented, highlighting existing differences. Finally, conservation laws for continua are presented, to end with a note on material frame indifference and frame invariance.

2.1.1. Description of motion

Continuum mechanics theory is used to describe the behavior of a body defined as *continuous*. A body is considered as a set of particles connected to each other, with no or a limited number of discontinuities. Due to the continuous nature of such a body, this one can be associated with a specific occupied domain in the Euclidean space, often referred to with the term *configuration*. Hence, a material point belonging to the body is related to a unique set of three real numbers (X_1, X_2, X_3).

To unequivocally describe the motion of a continuum, a certain configuration has to be assumed as *reference*. Let us consider a body that undergoes motion from an initial undeformed configuration, starting at the time t_0. After a certain time t, the body will occupy the *current configuration*, which is generally deformed, as illustrated in Figure 2.1.

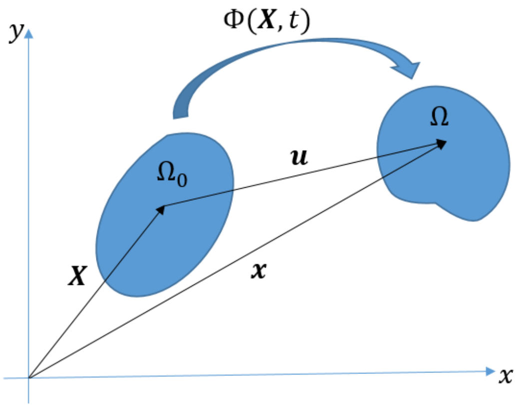


Figure 2.1: Initial (undeformed) and current (deformed) configurations

The position of a material particle in the reference configuration is described by the vector:

$$\mathbf{X} = \sum_{i=1}^{3} X_i \mathbf{e}_i \tag{2.1.1.1}$$

Where X_i are the components of the position vector and $\mathbf{e}_i$ are the base vectors of the Cartesian coordinate system. In the current configuration, the same particle will occupy a position $\mathbf{x}$, generally different from $\mathbf{X}$. The motion of the particle is a function of the particle material coordinates $\mathbf{X}$ and time t:

$$\mathbf{x} = \Phi(\mathbf{X}, t) \tag{2.1.1.2}$$

Where $\mathbf{x}$ is the position of the particle in the current configuration and Φ is the function that is mapping the initial configuration to the current configuration. The mapping function is assumed to be invertible and differentiable, so that it is possible to describe the position $\mathbf{X}$ in the reference configuration in terms of the current position $\mathbf{x}$ and time t:

$$\mathbf{X} = \Phi^{-1}(\mathbf{x}, t) \tag{2.1.1.3}$$

In continuum mechanics, two different approaches may be used to describe the motion of a body. In the first one, the *material* or *Lagrangian* description, the material coordinates X_i and the time t are taken as independent variables. In the second approach, the current coordinates x_i and the time t are used as independent variables and is given the name of *spatial* or *Eulerian description*. The Lagrangian description is following the material particle through its motion and is typically used when dealing with solids. It is indeed particularly useful to define an undeformed reference configuration, as the stresses in solids may depend on the deformation history or path, for example when damage or plasticity occurs [2]. On the other hand, with the Eulerian description, the current configuration coincides with the reference configuration for every t greater than t_0. The spatial description is preferred to describe the motion of fluids, since it is not really needed to define an initial configuration in this case. As the focus will be on solid mechanics, only the Lagrangian description will be used in this project.

The displacement of a certain material particle undergoing motion is given by the difference between current and reference position vectors:

$$\begin{aligned} \mathbf{u} &= \mathbf{x} - \mathbf{X} \\ &= \Phi(\mathbf{X}, t) - \mathbf{X} \end{aligned} \tag{2.1.1.4}$$

The velocity $\dot{\mathbf{u}}$ of the material particle is defined as the rate of change of its position vector. Therefore, it is obtained by derivation of the position vector with respect to time:

$$\dot{\mathbf{u}}(\mathbf{X},\mathrm{t}) = \frac{\partial \mathbf{u}(\mathbf{X},\mathrm{t})}{\partial \mathrm{t}} \tag{2.1.1.5}$$

In a similar fashion, the acceleration $\ddot{\mathbf{u}}$ of a certain particle is the rate of change of its velocity vector with respect to time. Hence, it is given by:

$$\ddot{\mathbf{u}}(\mathbf{X},\mathrm{t}) = \frac{\partial \dot{\mathbf{u}}(\mathbf{X},\mathrm{t})}{\partial \mathrm{t}} \tag{2.1.1.6}$$

2.1.2. Strain measures

Principal stretches can be intuitively used in order to quantify the amount of deformation to which a structure is subjected. However, in order to unequivocally establish constitutive equations in a geometric non-linear framework, it is necessary to adopt a specific *strain measure*.

Let us consider a particle indicated by the material point P, whose position is described in the reference configuration by the vector $\mathbf{X}$ as shown in Figure 2.2.

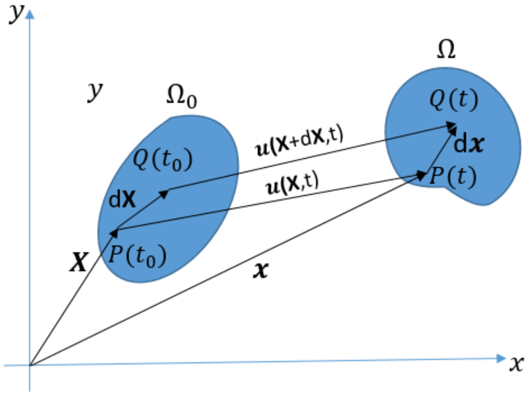


Figure 2.2: Motion of a continuum body

The same particle's position in the current configuration is given by the vector $\mathbf{x}$. Considering a neighbouring point Q located at the position $\mathbf{X} + \mathrm{d}\mathbf{X}$, the same point's position in the current configuration is given by the vector $\mathbf{x} + \mathrm{d}\mathbf{x}$. The current position vectors of the two points may also be written in terms of the displacement $\mathbf{u}$ as:

$$\mathbf{x} = \mathbf{X} + \mathbf{u}(\mathbf{X},\mathrm{t}) \tag{2.1.2.1}$$

$$\mathbf{x} + \mathrm{d}\mathbf{x} = \mathbf{X} + \mathrm{d}\mathbf{X} + \mathbf{u}(\mathbf{X} + \mathrm{d}\mathbf{X},\mathrm{t}) \tag{2.1.2.2}$$

Hence, the distance between the two points in the current configuration is obtained by subtracting the two vectors:

$$\mathrm{d}\mathbf{x} = \mathrm{d}\mathbf{X} + \mathbf{u}(\mathbf{X} + \mathrm{d}\mathbf{X},\mathrm{t}) - \mathbf{u}(\mathbf{X},\mathrm{t}) \tag{2.1.2.3}$$

Which may also be written using the definition of gradient of a vector as:

$$\mathrm{d}\mathbf{x} = \mathrm{d}\mathbf{X} + \nabla\mathbf{u} \cdot \mathrm{d}\mathbf{X} \tag{2.1.2.4}$$

This leads to the definition of the deformation gradient $\mathbf{F}$, which is defined as:

$$\mathbf{F}(\mathbf{X},\mathrm{t}) = \frac{\partial \mathbf{x}(\mathbf{X},\mathrm{t})}{\partial \mathbf{X}} \tag{2.1.2.5}$$

This one corresponds to the Jacobian matrix of the motion, which in three dimensions can be written in matrix form as:

$$\mathbf{F} = \begin{bmatrix} \frac{\partial x_1}{\partial X_1} & \frac{\partial x_1}{\partial X_2} & \frac{\partial x_1}{\partial X_3} \\ \frac{\partial x_2}{\partial X_1} & \frac{\partial x_2}{\partial X_2} & \frac{\partial x_2}{\partial X_3} \\ \frac{\partial x_3}{\partial X_1} & \frac{\partial x_3}{\partial X_2} & \frac{\partial x_3}{\partial X_3} \end{bmatrix} \tag{2.1.2.6}$$

The deformation gradient is an important variable that is used to describe the change in shape experienced by a solid body. This is also used for the definition of different strain measures in continuum mechanics theory. Indeed, the infinitesimal line segment $\mathrm{d}\mathbf{X}$ in the undeformed configuration is related to the line segment $\mathrm{d}\mathbf{x}$ in the deformed configuration through the deformation gradient:

$$\mathrm{d}\mathbf{x} = \mathbf{F} \cdot \mathrm{d}\mathbf{X} \qquad \text{or} \qquad \begin{bmatrix} \mathrm{d}x_1 \\ \mathrm{d}x_2 \\ \mathrm{d}x_3 \end{bmatrix} = \begin{bmatrix} 1 + \frac{\partial u_1}{\partial X_1} & \frac{\partial u_1}{\partial X_1} & \frac{\partial u_1}{\partial X_3} \\ \frac{\partial u_2}{\partial X_1} & 1 + \frac{\partial u_2}{\partial X_2} & \frac{\partial u_2}{\partial X_3} \\ \frac{\partial u_3}{\partial X_1} & \frac{\partial u_3}{\partial X_2} & 1 + \frac{\partial u_3}{\partial X_3} \end{bmatrix} \begin{bmatrix} \mathrm{d}X_1 \\ \mathrm{d}X_2 \\ \mathrm{d}X_3 \end{bmatrix} \tag{2.1.2.7}$$

Where the relation presented in Equation 2.1.2.4 was used in the derivation of the matrix form. Hence, the deformation gradient may be expressed as a function of the displacement gradient:

$$\mathbf{F} = \mathbf{I} + \nabla\mathbf{u} \tag{2.1.2.8}$$

The determinant of the deformation gradient J is a measure of the volume change caused by the deformation:

$$J = \det(\mathbf{F}) \tag{2.1.2.9}$$

The conservation of mass implies that the Jacobian must always be bigger than zero. The Jacobian is used to relate integrals in the current configuration Ω to integrals in the reference configuration Ω_0, according to the integration by substitution theorem of calculus:

$$\int_{\Omega} f \mathrm{d}\Omega = \int_{\Omega_0} f J \,\mathrm{d}\Omega_0 \tag{2.1.2.10}$$

Desirable characteristics of a strain measure are the following:

- It should be a continuous, monotonic and unique function of displacement gradients.
- It should be a symmetric second-order tensor.
- It must identically vanish in case of rigid body motion.
- It should reduce to the infinitesimal strain measure if the deformations become ”small”.

In particular, the first two attributes are convenient from a mathematical point of view. The ability to correctly represent a rigid body motion is a a crucial feature, since it is undesired to induce unrealistic non-zero stress in case of rigid body motion. On the other hand, the last feature is not strictly necessary, even though it is convenient for a smooth transition between the linear and non-linear regimes. Another important criterion for the validity of a strain measure is that it needs to satisfy the condition of objectivity, or *frame-indifference*. This one will be discussed more in detailed in Section 2.1.5. Given the notions presented so far, the most popular strain measures used in continuum mechanics can be introduced.

Green strain tensor

The Green strain tensor $\mathbf{E}$ measures the difference between the square of the segment length in the current configuration and the square of the same segment length in the reference configuration, and it is defined as:

$$d\mathbf{x}\cdot d\mathbf{x}-d\mathbf{X}\cdot d\mathbf{X}=2d\mathbf{X}\cdot\mathbf{E}\cdot d\mathbf{X} \tag{2.1.2.11}$$

Using the relation between deformed and undeformed segments expressed in Equation 2.1.5.4 and rearranging yields:

$$\mathbf{E}=\frac{1}{2}(\mathbf{F}^T\cdot\mathbf{F}-\mathbf{I}) \qquad \text{or} \qquad E_{ij}=\frac{1}{2}(F_{ki}F_{kj}-\delta_{ij}) \tag{2.1.2.12}$$

Alternatively, it is possible to express the Green strain tensor as a function of the displacement gradient through Equation 2.1.2.8:

$$\mathbf{E}=\frac{1}{2}(\nabla\mathbf{u}+\nabla\mathbf{u}^T+\nabla\mathbf{u}^T\cdot\nabla\mathbf{u}) \qquad \text{or} \qquad E_{ij}=\frac{1}{2}\left(\frac{\partial u_j}{\partial X_i}+\frac{\partial u_i}{\partial X_j}+\frac{\partial u_k}{\partial X_j}\frac{\partial u_k}{\partial X_i}\right) \tag{2.1.2.13}$$

Where the third term in Equation 2.1.2.13 is negligible when dealing with infinitesimal strains, since it represents the contribution of higher order terms. It is easy to verify that the Green strain is able to correctly represent rigid body motion, returning a zero strain.

Almansi strain tensor

The Almansi strain tensor $\mathbf{E}^*$ is obtained from the same equation of the Green strain tensor, expressing all the quantities as a function of the spatial coordinates:

$$d\mathbf{x}\cdot d\mathbf{x}-d\mathbf{X}\cdot d\mathbf{X}=2d\mathbf{x}\cdot\mathbf{E}^*\cdot d\mathbf{x} \tag{2.1.2.14}$$

Again, it is possible to express the strain tensor as a function of the deformation gradient:

$$\mathbf{E}^*=\frac{1}{2}\left(\mathbf{I}-(\mathbf{F}^{-1})^T\cdot\mathbf{F}^{-1}\right) \qquad \text{or} \qquad E^*_{ij}=\frac{1}{2}(\delta_{ij}-F^{-1}_{ki}F^{-1}_{kj}) \tag{2.1.2.15}$$

The Almansi strain tensor also satisfies the condition of zero strain in case of rigid body motion. It is worth mentioning that, assuming infinitesimal strains, the Almansi strain tensor is equivalent to the Green strain tensor, as the higher order terms can be neglected. However, the two strain tensors are different in geometrically non-linear problems, where the higher order terms are not negligible.

Rate of deformation tensor

The rate of deformation tensor $\mathbf{D}$ is an alternative tool to measure deformation in a body. Let us first introduce the velocity gradient $\mathbf{L}$, defined as the spatial variation of the velocity vector:

$$\mathbf{L}=\frac{\partial\dot{\mathbf{u}}}{\partial\mathbf{x}}=\nabla\dot{\mathbf{u}} \qquad \text{or} \qquad L_{ij}=\frac{\partial\dot{u}_i}{\partial x_j} \tag{2.1.2.16}$$

In particular, the symmetric/anti-symmetric decomposition can be applied to the velocity gradient, which yields:

$$\mathbf{L}=\frac{1}{2}(\mathbf{L}+\mathbf{L}^T)+\frac{1}{2}(\mathbf{L}-\mathbf{L}^T) \qquad \text{or} \qquad L_{ij}=\frac{1}{2}(L_{ij}+L_{ji})+\frac{1}{2}(L_{ij}-L_{ji}) \tag{2.1.2.17}$$

The rate of deformation tensor $\mathbf{D}$ is then defined as the symmetric part of the decomposition, while its skew-symmetric part $\mathbf{W}$ is called *spin tensor*:

$$\mathbf{D} = \frac{1}{2}(\mathbf{L} + \mathbf{L}^{\mathrm{T}}) \quad \text{or} \quad \mathrm{D}_{\mathrm{ij}} = \frac{1}{2}\left(\frac{\partial \dot{\mathrm{u}}_{\mathrm{i}}}{\partial \mathrm{x}_{\mathrm{j}}} + \frac{\partial \dot{\mathrm{u}}_{\mathrm{j}}}{\partial \mathrm{x}_{\mathrm{i}}}\right) \tag{2.1.2.18}$$

$$\mathbf{W} = \frac{1}{2}(\mathbf{L} - \mathbf{L}^{\mathrm{T}}) \quad \text{or} \quad \mathrm{W}_{\mathrm{ij}} = \frac{1}{2}\left(\frac{\partial \dot{\mathrm{u}}_{\mathrm{i}}}{\partial \mathrm{x}_{\mathrm{j}}} - \frac{\partial \dot{\mathrm{u}}_{\mathrm{j}}}{\partial \mathrm{x}_{\mathrm{i}}}\right) \tag{2.1.2.19}$$

The rate of deformation tensor $\mathbf{D}$ measures the rate of stretching of material fibers in a deformed solid, while the spin tensor $\mathbf{W}$ quantifies the angular velocity of material fibers passing through a material point. Alternatively, the rate of deformation tensor can be expressed as a function of the deformation gradient $\mathbf{F}$ as:

$$\mathbf{D} = \frac{1}{2}(\dot{\mathbf{F}} \cdot \mathbf{F}^{-1} + \mathbf{F}^{-\mathrm{T}} \cdot \dot{\mathbf{F}}^{\mathrm{T}}) \tag{2.1.2.20}$$

2.1.3. Stress measures

In the study of continua, it is of main interest to determine how the forces are transmitted through a medium. Considering a continuous body, the force distribution is represented through *surface forces* and *body forces*. Surface forces are those forces applied to the external (real) or internal (imaginary) boundaries of the body. On the other hand, body forces are acting throughout a volume, as an example gravity or electromagnetic forces can be mentioned. In continuum mechanics, the concept of *stress* is introduced to describe how forces are transmitted through a body. In this section, the most common stress definitions are presented, highlighting their differences.

Cauchy stress

Let us consider a body as the one shown in Figure 2.3, which is cut in two parts by a plane passing through an arbitrary point P. The plane is characterized by the outer normal vector $\mathbf{n}$. The body is assumed to be in equilibrium upon application of the forces $\mathbf{F}_1$, $\mathbf{F}_2$, $\mathbf{F}_3$ and $\mathbf{F}_4$.

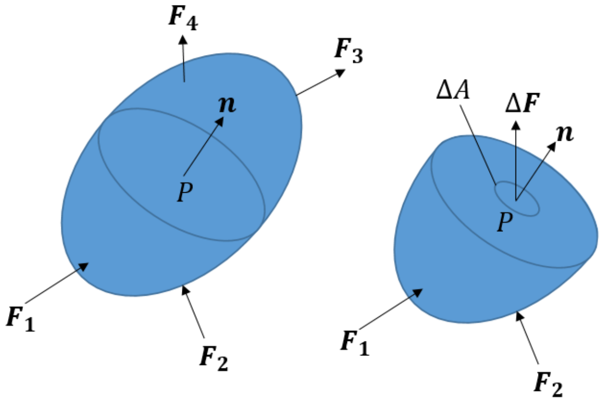


Figure 2.3: Visualization of the stress vector

Considering the bottom portion, an internal force $\Delta\mathbf{F}$, acting on a portion of area $\Delta\mathrm{A}$, will be present in order to maintain equilibrium with the forces $\mathbf{F}_1$ and $\mathbf{F}_2$. This leads to the introduction of a *stress vector* at the point P, which is defined as:

$$\mathbf{t}_{\mathrm{n}} = \lim_{\Delta\mathrm{A}\to 0} \frac{\Delta\mathbf{F}}{\Delta\mathrm{A}} \tag{2.1.3.1}$$

Where the subscript n refers to the fact that the force is acting on a surface with outward normal vector $\mathbf{n}$. According to the third Newton's law on action and reaction, there must be a stress vector

$\mathbf{t}_{-\mathrm{n}}$ at the same point on the top portion of the body, which is equal in magnitude but with an opposite direction with respect to $\mathbf{n}$:

$$\mathbf{t}_{\mathrm{n}} = -\mathbf{t}_{-\mathrm{n}} \tag{2.1.3.2}$$

The stress vector is a function of the position $\mathbf{x}$, time t and the surface in terms of outward normal vector $\mathbf{n}$:

$$\mathbf{t}_{\mathrm{n}} = \mathbf{t}(\mathbf{x}, \mathrm{t}, \mathbf{n}) \tag{2.1.3.3}$$

In particular, the dependence on the outward normal vector can be expressed as follows:

$$\mathbf{t}_{\mathrm{n}} = \mathbf{n} \cdot \boldsymbol{\sigma}(\mathbf{x}, \mathrm{t}) \tag{2.1.3.4}$$

Where $\boldsymbol{\sigma}$ is called the *Cauchy stress tensor*. The Cauchy stress tensor completely describes the forces acting on a deformed solid. Considering a basis of vectors $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$ in the $3\mathrm{D}$ Euclidean space, and denoting with $\mathrm{t}_\mathrm{i}(\mathbf{e}_\mathrm{j})$ the surface traction vector component in the j direction, acting on the plane with normal in the i direction, the Cauchy stress tensor may be written in matrix form as:

$$\boldsymbol{\sigma} = \begin{bmatrix} \mathrm{t}_1(\mathbf{e}_1) & \mathrm{t}_1(\mathbf{e}_2) & \mathrm{t}_1(\mathbf{e}_3) \\ \mathrm{t}_2(\mathbf{e}_1) & \mathrm{t}_2(\mathbf{e}_2) & \mathrm{t}_2(\mathbf{e}_3) \\ \mathrm{t}_3(\mathbf{e}_1) & \mathrm{t}_3(\mathbf{e}_2) & \mathrm{t}_3(\mathbf{e}_3) \end{bmatrix} \tag{2.1.3.5}$$

The Cauchy stress tensor represents the components of the force per unit area of the deformed solid. For this reason it is also called the *true stress* and is the most used stress measure. It can be proved that, imposing the conservation of angular momentum, the Cauchy stress tensor turns out to be symmetric.

Nominal (First Piola-Kirchhoff) stress

The Nominal or First Piola-Kirchhoff (PK1) stress $\mathbf{P}$ is defined in a fashion similar to the Cauchy stress, with the exception that this one is defined in terms of the area and the normal of the undeformed configuration. Denoting with $\mathrm{d}\mathbf{f}$ the force acting on the deformed configuration, the following relation exists between the force, the surface traction $\mathbf{t}$ and the Cauchy stress tensor $\boldsymbol{\sigma}$:

$$\mathrm{d}\mathbf{f} = \mathbf{t}\mathrm{dA} = \mathbf{n} \cdot \boldsymbol{\sigma}\mathrm{dA} \tag{2.1.3.6}$$

Where dA is the deformed area of the element in the current configuration and $\mathbf{n}$ is the vector normal to the deformed area. The force acting on the deformed area is related to the PK1 stress through the following:

$$\mathrm{d}\mathbf{f} = \mathbf{t}_0\mathrm{dA}_0 = \mathbf{n}_0 \cdot \mathbf{P}\mathrm{dA}_0 \tag{2.1.3.7}$$

Where dA_0 is the undeformed area of the element in the reference configuration and $\mathbf{n}_0$ is the vector normal to the undeformed area. Due to the fact that this stress tensor measures the force acting on the deformed configuration in terms of the area in the undeformed configuration, it does not have any specific physical meaning. On the other hand, there may be situations in which it is more convenient to express the equations of motion with respect to the reference configuration rather than in the current one.

To express the PK1 stress in terms of the Cauchy stress tensor we need to introduce the Nanson's relation, which relates the current normal to the reference normal. This one is given as follows:

$$\mathbf{n} \cdot \mathrm{dA} = \mathrm{J}\mathbf{n}_0 \cdot \mathbf{F}^{-1}\mathrm{dA}_0 \tag{2.1.3.8}$$

Hence, using the latter, together with 2.1.3.6 and 2.1.3.7, yields:

$$\mathbf{n} \cdot \boldsymbol{\sigma}\mathrm{dA} = \mathbf{n}_0 \cdot \mathbf{P}\mathrm{dA}_0 \qquad \rightarrow \qquad \mathbf{P} = \mathrm{J}\mathbf{F}^{-1} \cdot \boldsymbol{\sigma} \tag{2.1.3.9}$$

It is worth to note from Equation 2.1.3.9 that the nominal stress tensor is generally not symmetric.

Material (Second Piola-Kirchhoff) stress

The Material or Second Piola-Kirchhoff (PK2) stress tensor S is measuring the force $\mathrm{d}\tilde{\mathbf{f}}$ acting on the undeformed body per unit of undeformed area. The force acting on the body in the reference configuration is related to the one acting on the body in the current configuration $\mathrm{d}\mathbf{f}$ through the deformation gradient $\mathbf{F}$:

$$\mathrm{d}\mathbf{f} = \mathbf{F} \cdot \mathrm{d}\tilde{\mathbf{f}} \tag{2.1.3.10}$$

At the same time, the following relation exists between the force $\mathrm{d}\tilde{\mathbf{f}}$ and the PK2 stress tensor:

$$\mathrm{d}\tilde{\mathbf{f}} = \mathbf{n_0} \cdot \mathbf{S}\mathrm{dA_0} \tag{2.1.3.11}$$

Using Equation 2.1.3.10 and the relation found in Equation 2.1.3.9, the material stress tensor S can be expressed in terms of the Cauchy stress tensor $\boldsymbol{\sigma}$:

$$\mathbf{F}^{-1} \cdot \mathrm{d}\mathbf{f} = \mathbf{n_0} \cdot \mathbf{S}\mathrm{dA_0} \qquad \rightarrow \qquad \mathbf{S} = \mathrm{J}\mathbf{F}^{-1} \cdot \boldsymbol{\sigma} \cdot \mathbf{F}^{-\mathrm{T}} \tag{2.1.3.12}$$

The PK2 stress tensor is therefore symmetric.

Co-rotational Cauchy stress tensor

This stress is equivalent to the Cauchy stress tensor expressed in terms of components referred to a coordinate system which rotates with the material but does not deform, given the name of *co-rotational system*. The CR system is constructed at every material point and is defined by the base vectors $\hat{\mathrm{e}}_\mathrm{i}$, as shown in Figure 2.4

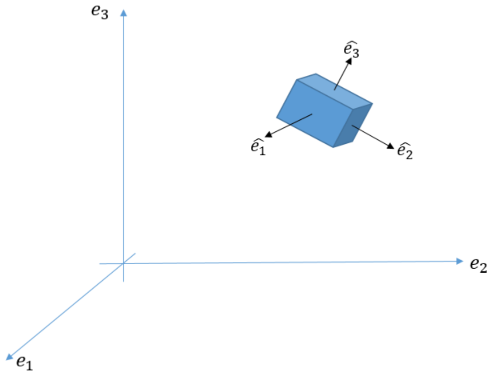


Figure 2.4: Visualization of the co-rotational reference system

A crucial aspect is to correctly define the rotation of the material in order to obtain such a coordinate system. To this regard, the polar decomposition theorem is typically used, according to which the deformation gradient can be decomposed into a product between a rotation $\mathbf{R}$ and right stretch tensor $\mathbf{U}$ [2]:

$$\mathbf{F} = \mathbf{R} \cdot \mathbf{U} \qquad \text{or} \qquad \mathrm{F_{ij}} = \mathrm{R_{ik}U_{kj}} \tag{2.1.3.13}$$

Where $\mathbf{R}$ is an orthogonal matrix and $\mathbf{U}$ is symmetric positive definite, so they are characterized by the following properties:

$$\mathbf{R}^{-1} = \mathbf{R}^{\mathrm{T}} \qquad \text{and} \qquad \mathbf{U} = \mathbf{U}^{\mathrm{T}} \tag{2.1.3.14}$$

This theorem proves that the deformation gradient consists in a component that stretches the body along a set of orthogonal axes, represented by the symmetric positive definite mapping $\mathbf{U}$, and a rigid body rotation represented by the orthogonal tensor $\mathbf{R}$. Equivalently, it is proved that the deformation gradient can be decomposed into a product of a left stretch tensor $\mathbf{V}$ and a rotation tensor $\mathbf{R}$:

$$\mathbf{F} = \mathbf{V} \cdot \mathbf{R} \qquad \text{or} \qquad \mathrm{F_{ij}} = \mathrm{V_{ik}R_{kj}} \tag{2.1.3.15}$$

Once the material rotation is determined, the co-rotational Cauchy stress tensor is given by:

$$\hat{\boldsymbol{\sigma}} = \mathbf{R}^{\mathrm{T}} \cdot \boldsymbol{\sigma} \cdot \mathbf{R} \qquad \text{or} \qquad \hat{\sigma}_{\mathrm{ij}} = \mathrm{R}^{\mathrm{T}}_{\mathrm{ik}}\sigma_{\mathrm{kl}}\mathrm{R}_{\mathrm{lj}} \tag{2.1.3.16}$$

Using the co-rotational stress $\hat{\boldsymbol{\sigma}}$ is convenient, as it provides the correct physical components of stress in the portion of material that is considered. In a similar fashion, strain measures can also be expressed in terms of co-rotational coordinates through the same transformation given in Equation 2.1.3.16.

For convenience, the transformation equations between different stress measures are reported in Table 2.1.

Transformations between stress measures			
Cauchy stress $\boldsymbol{\sigma}$	Nominal stress $\mathbf{P}$	2nd Piola-Kirchhoff stress $\mathbf{S}$	Co-rotational Cauchy stress $\hat{\boldsymbol{\sigma}}$
$\boldsymbol{\sigma}$	$\boldsymbol{\sigma} = \mathrm{J}^{-1}\mathbf{F} \cdot \mathbf{P}$	$\boldsymbol{\sigma} = \mathrm{J}^{-1}\mathbf{F} \cdot \mathbf{S} \cdot \mathbf{F}^{\mathrm{T}}$	$\boldsymbol{\sigma} = \mathbf{R} \cdot \hat{\boldsymbol{\sigma}} \cdot \mathbf{R}^{\mathrm{T}}$
$\mathbf{P} = \mathrm{J}\mathbf{F}^{-1} \cdot \boldsymbol{\sigma}$	$\mathbf{P}$	$\mathbf{P} = \mathbf{S} \cdot \mathbf{F}^{\mathrm{T}}$	$\mathbf{P} = \mathrm{J}\mathbf{U}^{-1} \cdot \hat{\boldsymbol{\sigma}} \cdot \mathbf{R}^{\mathrm{T}}$
$\mathbf{S} = \mathrm{J}\mathbf{F}^{-1} \cdot \boldsymbol{\sigma} \cdot \mathbf{F}^{-\mathrm{T}}$	$\mathbf{S} = \mathbf{P} \cdot \mathbf{F}^{-\mathrm{T}}$	$\mathbf{S}$	$\mathbf{S} = \mathrm{J}\mathbf{U}^{-1} \cdot \hat{\boldsymbol{\sigma}} \cdot \mathbf{U}^{-1}$
$\hat{\boldsymbol{\sigma}} = \mathbf{R}^{\mathrm{T}} \cdot \boldsymbol{\sigma} \cdot \mathbf{R}$	$\hat{\boldsymbol{\sigma}} = \mathrm{J}^{-1}\mathbf{U} \cdot \mathbf{P} \cdot \mathbf{R}$	$\hat{\boldsymbol{\sigma}} = \mathrm{J}^{-1}\mathbf{U} \cdot \mathbf{S} \cdot \mathbf{U}$	$\hat{\boldsymbol{\sigma}}$

Table 2.1: Relations between different stress measures

These ones, together with the previously introduced strain measures, are used to formulate the governing equations of a continuum.

2.1.4. Conservation laws

Conservation laws for continua consist in a set of fundamental equations that must be satisfied by every physical system. When these laws are expressed in the form of differential equations, they are referred to with the name of *local principles* or *strong form*. On the other hand, when these ones are presented in the integral form, they are called *global principles* or *weak form*. Although it is mathematically proven that the two forms are equivalent, the second form is referred as weak due to the fact that the smoothness requirements on the solution are relaxed. In this section, only the strong form of the laws is reported, omitting their detailed derivation for brevity.

Conservation of mass

Considering a body occupying a specific material domain Ω, its mass is defined as:

$$\mathrm{m}(\Omega) = \int_{\Omega} \rho(\mathbf{x}, \mathrm{t})\, \mathrm{d}\Omega \tag{2.1.4.1}$$

Where $\rho(\mathbf{x}, t)$ is the density of the body, which in general is a function of position and time. The mass conservation principle states that the mass belonging to a material domain must not change in time.

Denoting with Ω_0 the material domain occupied by the body in the reference configuration:

$$\int_{\Omega} \rho(\mathbf{X}, t)\, d\Omega = \int_{\Omega_0} \rho_0(\mathbf{X})\, d\Omega_0 \tag{2.1.4.2}$$

Where ρ_0 is the density of the body at the reference configuration, which is only a function of the material coordinates. Using the Jacobian determinant, it is possible to relate the integral in the current configuration to the one in the reference configuration:

$$\int_{\Omega} \rho\, d\Omega = \int_{\Omega_0} \rho J\, d\Omega_0 \qquad \rightarrow \qquad \int_{\Omega_0} (\rho J - \rho_0)\, d\Omega_0 \tag{2.1.4.3}$$

Which, assuming the integrand to be smooth, is equal to:

$$\rho(\mathbf{X}, t) J(\mathbf{X}, t) = \rho_0(\mathbf{X}) \tag{2.1.4.4}$$

Equation 2.1.4.4 represents the strong form of the principle of mass conservation.

Conservation of linear momentum

The conservation of linear momentum is the equivalent of Newton's second law, which relates the forces acting on a body to its acceleration. According to Newton's second law, the material time derivative of the linear momentum $\mathbf{p}$ has to be equal to the sum of the forces $\mathbf{f}$ acting on the body:

$$\frac{d\mathbf{p}}{dt} = \mathbf{f} \tag{2.1.4.5}$$

Adopting an Eulerian description, denoting with $\mathbf{b}$ the body forces acting on the body, elaborating on Equation 2.1.4.5 yields the following:

$$\rho \frac{\partial \dot{\mathbf{u}}}{\partial t} = \rho \mathbf{b} + \nabla \cdot \boldsymbol{\sigma} \tag{2.1.4.6}$$

Equation 2.1.4.6 represents the strong form of the conservation of linear momentum. This one may also be formulated adopting a Lagrangian description. Doing so, the conservation of linear momentum is reduced to the following partial differential equation:

$$\rho_0 \frac{\partial \dot{\mathbf{u}}}{\partial t} = \rho_0 \mathbf{b} + \nabla_0 \cdot \mathbf{P} \tag{2.1.4.7}$$

Conservation of the angular momentum

The principle of conservation of angular momentum states that the sum of the moments of all forces acting on a body about a fixed point has to be equal to the rate of change of angular momentum about the same point. Denoting with $\mathbf{t}$ the surface forces acting on the boundaries of the body Γ and with $\mathbf{b}$ the body forces acting on the volume of the body Ω, the integral form of the principle is given by:

$$\frac{d}{dt} \int_{\Omega} \mathbf{x} \times \rho \dot{\mathbf{u}}\, d\Omega = \int_{\Omega} \mathbf{x} \times \rho \mathbf{b}\, d\Omega + \int_{\Gamma} \mathbf{x} \times \mathbf{t}\, d\Gamma \tag{2.1.4.8}$$

Carrying out the derivation, associating the surface traction to the Cauchy stress tensor, the conservation of the angular momentum requires the Cauchy stress tensor to be symmetric:

$$\sigma_{12} - \sigma_{21} = 0; \qquad \sigma_{23} - \sigma_{32} = 0; \qquad \sigma_{31} - \sigma_{13} = 0; \qquad \rightarrow \qquad \boldsymbol{\sigma} = \boldsymbol{\sigma}^T \tag{2.1.4.9}$$

On the other hand, adopting a material description and using the relations between the Cauchy, the PK1 and PK2 stress reported in Table 2.1, it can be proved that the principle reduces to the symmetry requirement for the PK2 stress tensor:

$$\mathbf{S} = \mathbf{S}^T \tag{2.1.4.10}$$

Conservation of energy

The principle of conservation of energy states that in a system, in which the only sources of energy are mechanical work and heat, the rate of change of kinetic and internal energy in a system has to be equal to the sum of the rate of work done by external forces plus the rate of work coming from the heat flux or other heat supplies:

$$\frac{\mathrm{d}}{\mathrm{dt}}\int_{\Omega}\left(\frac{1}{2}\rho\dot{\mathbf{u}}\cdot\dot{\mathbf{u}}+\rho\mathrm{w}_{\mathrm{int}}\right)\mathrm{d}\Omega=\int_{\Gamma}\mathbf{t}\cdot\dot{\mathbf{u}}\,\mathrm{d}\Gamma+\int_{\Omega}\rho\mathbf{b}\cdot\dot{\mathbf{u}}\,\mathrm{d}\Omega-\int_{\Gamma}\mathbf{n}\cdot\mathbf{q}\,\mathrm{d}\Gamma+\int_{\Omega}\rho\mathrm{q}_{\mathrm{s}}\,\mathrm{d}\Omega \tag{2.1.4.11}$$

Where $\mathrm{w}_{\mathrm{int}}$ is the internal energy of the system per unit mass, $\mathbf{q}$ is the heat flux vector and q_{s} is the power supplied by the heat sources per unit mass. It is worth noting that the heat flux contribution is characterized by a negative sign in front of it, according to the classical sign convention a positive heat flux is transferred to a system. The principle of conservation of energy is also known as the *first law of thermodynamics*. Developing Equation 2.1.4.11, the strong form of the principle of conservation of energy is given by:

$$\rho\frac{\mathrm{dw}_{\mathrm{int}}}{\mathrm{dt}}=\boldsymbol{\sigma}:\mathbf{D}-\nabla\cdot\mathbf{q}+\rho\mathrm{q}_{\mathrm{s}}=0 \tag{2.1.4.12}$$

On the other hand, using material coordinates, the principle of conservation of energy becomes:

$$\rho_0\frac{\mathrm{dw}_{\mathrm{int}}}{\mathrm{dt}}=\dot{\mathbf{F}}^{\mathrm{T}}:\mathbf{P}-\nabla_0\cdot(\mathbf{F}^{-1}\cdot\mathbf{q})+\rho_0\mathrm{q}_{\mathrm{s}} \tag{2.1.4.13}$$

By comparison of Equation 2.1.4.12 with Equation 2.1.4.13, let us consider the case in which only mechanical energies are involved. In such circumstances, it is worth noting that different strain/stress measures are used in order to calculate the internal power. The internal power is defined as the rate of work involved in order to change the shape of a unit volume of material. In particular, the Cauchy stress $\boldsymbol{\sigma}$ is conjugate in power to the rate of deformation $\mathbf{D}$, while the nominal stress $\mathbf{P}$ is conjugate in power to the material time derivative of the deformation gradient. In a similar fashion it can be proved that the PK2 stress $\mathbf{S}$ is conjugate in power to the time derivative of the Green strain tensor $\dot{\mathbf{E}}$. Power conjugates are used in the formulation of the weak forms of the conservation laws, which are the starting point for the FE formulation.

Second law of thermodynamics

Let us denote with $\mathrm{s}(\mathbf{x},\mathrm{t})$ the specific entropy of a continuum body and with ρ its density. The second law of thermodynamics states that the entropy of a system can only increase over time, and it is always greater or equal to the entropy inflow across its boundary surface plus the entropy supply throughout the volume:

$$\rho\frac{\mathrm{ds}}{\mathrm{dt}}\geq-\nabla\left(\frac{\mathbf{q}}{\Theta}\right)+\frac{\rho\mathrm{q}_{\mathrm{s}}}{\Theta} \tag{2.1.4.14}$$

Where $\mathbf{q}$ is the heat flux vector, q_{s} is the power supplied by heat sources per unit mass and Θ is the absolute temperature. This is known as the *second law of thermodynamics*, and it is also called *Clausius inequality*. In particular, it can be proven that the first law of thermodynamics implies the second law and vice versa. Although the second law of thermodynamics might seem to be quite irrelevant to this project, it has an important implication when it comes to fracture mechanics. As a matter of fact, the entropy inequality implies that when a structure undergoes damage, it is not possible to observe healing further in time: the damage can either grow or keep stable.

These conservation principles constitute the necessary foundations to unequivocally describe the motion of a continuum body. As previously mentioned, different strain and stress measures may be used for this purpose. Before delving into the FE formulation, the validity of such measures has to be considered, especially when these are used to specify the material constitutive relation.

2.1.5. Material frame indifference and frame invariance

The principle of *material frame indifference* or *objectivity* states that the material response description has to be independent of the observer. Objectivity is crucial, since the validity of the physical laws describing a body must always hold, regardless of the adopted reference system. For the sake of this project, it is of main interest how the involved physical quantities are behaving under a change of reference frame.

Let us consider a set of physical quantities described in an inertial reference frame. It is here desired to express these quantities in an observer's reference frame, which translates and rotates with respect to the inertial reference frame. Denoting with a star superscript the quantities in the observer's reference frame and using no superscript for the quantities defined in the inertial reference frame, the motion in the observer's frame is related to the one in the inertial frame by the following:

$$\mathbf{x}^*(\mathrm{t}) = \mathbf{Q}(\mathrm{t}) \cdot \mathbf{x}(\mathrm{t}) + \mathbf{c}(\mathrm{t}) \qquad \text{with} \qquad \mathbf{Q}^{-1} = \mathbf{Q}^{\mathrm{T}} \tag{2.1.5.1}$$

Where $\mathbf{Q}(\mathrm{t})$ is an orthogonal tensor representing the rigid rotation between the two frames and $\mathbf{c}(\mathrm{t})$ the vector that represents the translation between the two. In particular, it is assumed that $\mathbf{Q}(0) = \mathbf{I}$ and $\mathbf{c}(0) = 0$. In order to preserve objectivity, it is important that all physical quantities are measured in a consistent way and that they do transform in a particular manner. Specifically:

- A scalar quantity, such as temperature or density, is objective only if it has the same value to all observers. In this case the quantity is called *frame invariant*.
- A vector quantity $\mathbf{u}$, such as a line element or the normal to a surface, must transform according to:

  $$\mathbf{u}^* = \mathbf{Q} \cdot \mathbf{u} \tag{2.1.5.2}$$

  In such a case, the vector quantity is said to be *objective* or *frame indifferent*.
- A tensor quantity, that is mapping an objective vector quantity onto another objective vector quantity is also said to be objective or frame indifferent. Such tensor has to transform according to:

  $$\boldsymbol{\sigma}^* = \mathbf{Q} \cdot \boldsymbol{\sigma} \cdot \mathbf{Q}^{\mathrm{T}} \tag{2.1.5.3}$$

To this regard, it is worth categorizing the previously introduced strains and stress measures. In particular:

- The deformation gradient is not frame indifferent, and transforms as follows:

  $$\mathbf{F}^* = \mathbf{Q} \cdot \mathbf{F} \tag{2.1.5.4}$$

- The Green strain tensor is frame invariant. Indeed, applying the transformation given by Equation 2.1.5.4, and due to the fact that the rotation tensor $\mathbf{Q}$ is orthogonal:

  $$\begin{aligned} \mathbf{E}^* &= \frac{1}{2}(\mathbf{F}^{*\mathrm{T}} \cdot \mathbf{F}^* - \mathbf{I}) \\ &= \frac{1}{2}(\mathbf{F}^{\mathrm{T}} \cdot \mathbf{Q}^{\mathrm{T}} \cdot \mathbf{Q} \cdot \mathbf{F} - \mathbf{I}) \\ &= \mathbf{E} \end{aligned} \tag{2.1.5.5}$$

- The rate of deformation tensor $\mathbf{D}$ is frame indifferent and transforms according to Equation 2.1.5.3.
- The Cauchy stress tensor $\boldsymbol{\sigma}$ is frame indifferent and transforms according to Equation 2.1.5.3.
- The PK1 stress tensor $\mathbf{P}$ is not frame indifferent, and it transforms as follows:

  $$\begin{aligned} \mathbf{P}^* &= \mathrm{J}\mathbf{F}^{*-1} \cdot \boldsymbol{\sigma}^* \\ &= \mathrm{J}(\mathbf{Q} \cdot \mathbf{F})^{-1} \cdot \mathbf{Q} \cdot \boldsymbol{\sigma} \cdot \mathbf{Q}^{\mathrm{T}} \\ &= \mathbf{P} \cdot \mathbf{Q}^{\mathrm{T}} \end{aligned} \tag{2.1.5.6}$$

- The PK2 stress tensor $\mathbf{S}$ is frame invariant, so that:

$$\mathbf{S}^* = \mathbf{S} \tag{2.1.5.7}$$

- The co-rotational Cauchy stress tensor $\hat{\boldsymbol{\sigma}}$ is frame invariant, as it can be proved that:

$$\hat{\boldsymbol{\sigma}} = \hat{\boldsymbol{\sigma}}^* \tag{2.1.5.8}$$

As previously mentioned, objectivity is quite important when it comes to define the constitutive laws for a material. These ones have to hold their validity under a change of reference. Hence, only objective strain and stress tensors should be used to define constitutive relations.

Having this necessary background defined, it is possible to dig into the finite element formulation, which is reported in the next section.

2.2. The Finite Element Method

To determine the motion of a continuum, the previously presented governing equations have to be solved. As previously mentioned, these can be formulated referring the physical quantities to either the material or spatial coordinates. When it comes to solid mechanics, a material description of the motion is preferred. In such a way, the equations are evaluated always at the same material points, which is especially useful for history-dependent materials. In the FE discretization for solid mechanics two different approaches are mainly used: the *Total Lagrangian* (TL) and the *Updated Lagrangian* (UL) formulations. In the TL formulation, all the quantities are referred to the reference configuration at time $\mathrm{t}_0 = 0$, while in the UL formulation the quantities are referred to the last converged configuration, which is assumed as the new reference configuration. The governing equations for the two approaches are summarized in Table 2.2 for convenience.

Governing equations	**Total Lagrangian**	**Updated Lagrangian**
Conservation of mass	$\rho(\mathbf{X},\mathrm{t})\mathrm{J}(\mathbf{X},\mathrm{t}) = \rho_0(\mathbf{X})$	$\rho(\mathbf{X},\mathrm{t})\mathrm{J}(\mathbf{X},\mathrm{t}) = \rho_0(\mathbf{X})$
Conservation of linear momentum	$\rho_0 \frac{\partial \dot{\mathbf{u}}}{\partial \mathrm{t}} = \rho_0 \mathbf{b} + \nabla_0 \cdot \mathbf{P}$	$\rho \frac{\partial \dot{\mathbf{u}}}{\partial \mathrm{t}} = \rho \mathbf{b} + \nabla \cdot \boldsymbol{\sigma}$
Conservation of angular momentum	$\mathbf{F} \cdot \mathbf{P} = \mathbf{P}^{\mathrm{T}} \cdot \mathbf{F}^{\mathrm{T}}$	$\boldsymbol{\sigma} = \boldsymbol{\sigma}^{\mathrm{T}}$
Conservation of energy	$\rho_0 \frac{\mathrm{dw_{int}}}{\mathrm{dt}} - \dot{\mathbf{F}}^{\mathrm{T}} : \mathbf{P} + \nabla_0 \cdot (\mathbf{F}^{-1} \cdot \mathbf{q}) - \rho_0 \mathrm{q_s} = 0$	$\rho \frac{\mathrm{dw_{int}}}{\mathrm{dt}} - \boldsymbol{\sigma} : \mathbf{D} + \nabla \cdot \mathbf{q} - \rho \mathrm{q_s} = 0$
Boundary conditions	$\mathbf{e}_\mathrm{i} \cdot \mathbf{n}_0 \cdot \mathbf{P} = \mathbf{e}_\mathrm{i} \cdot \bar{\mathbf{t}}_0$ on $\Gamma_{0\mathrm{t,i}}$ or $\mathbf{u} = \bar{\mathbf{u}}$ on $\Gamma_{0\mathrm{u}}$	$\mathbf{e}_\mathrm{i} \cdot \mathbf{n} \cdot \boldsymbol{\sigma} = \mathbf{e}_\mathrm{i} \cdot \bar{\mathbf{t}}$ on $\Gamma_{\mathrm{t,i}}$ or $\mathbf{u} = \bar{\mathbf{u}}$ on Γ_{u}
Initial conditions	$\dot{\mathbf{u}}(\mathbf{X},0) = \dot{\mathbf{u}}_0(\mathbf{X})$ and $\mathbf{P}(\mathbf{X},0) = \mathbf{P}_0(\mathbf{X})$	$\mathbf{u}(\mathbf{X},0) = \mathbf{u}_0(\mathbf{X})$ and $\boldsymbol{\sigma}(\mathbf{X},0) = \boldsymbol{\sigma}_0(\mathbf{X})$ or $\mathbf{u}(\mathbf{X},0) = \mathbf{u}_0(\mathbf{X})$ and $\dot{\mathbf{u}}(\mathbf{X},0) = \dot{\mathbf{u}}_0(\mathbf{X})$
Traction continuity	$[\![\mathbf{n}_0 \cdot \mathbf{P}]\!] = 0$ on $\Gamma_{0\mathrm{int}}$	$[\![\mathbf{n} \cdot \boldsymbol{\sigma}]\!] = 0$ on Γ_{int}

Table 2.2: Strong form of the governing equations for TL and UL approaches

The mass and energy conservation are scalar equations, assuming no transfer of heat. On the other hand, the conservation of linear momentum consists in a system of partial differential equations, whose number depends on the problem dimension $\mathrm{n_{dim}}$. The conservation of angular momentum does not contribute in the number of equations characterizing the motion, as it is only translating into specific conditions on the involved stress tensor. The constitutive material relation is used to couple the stress to the strain measures. Overall, the constitutive relation corresponds to $\mathrm{n}_{\mathrm{const}}$ equations, with $\mathrm{n}_{\mathrm{const}}$ given by:

$$\mathrm{n}_{\mathrm{const}} = \mathrm{n}_{\mathrm{dim}}(\mathrm{n}_{\mathrm{dim}} + 1)/2 \tag{2.2.0.1}$$

In addition to these, there is an additional equal number $\mathrm{n}_{\mathrm{const}}$ of compatibility equations, used to relate the strain measure to the displacements. Hence, to determine the motion, a total number of $2\mathrm{n}_{\mathrm{const}} + \mathrm{n}_{\mathrm{dim}} + 1$ equations must be solved.

Boundary conditions must be specified to solve the system of equations. Either the stress components or the displacement are prescribed on the boundary of the body. In a similar fashion, initial conditions can be imposed either on displacements and stresses or on displacements and velocities. The first ones are more common in solid mechanics problems due to the fact that, in most cases, it is easier to estimate the initial stresses in a body rather than the initial displacements.

A closed form solution to this system of partial differential equations can be obtained only for simple initial boundary conditions and geometrically well defined problems. As a result, approximate methods such as the Finite Element Method have been developed to obtain solutions for more complex cases. The Finite Element Method is derived by formulating the governing equations in a *weak form*, which is easier to solve from a computational point of view.

The steps involved in a FE analysis are the following:

- Discretization of the physical domain into elements and nodes.
- Assumption of a displacement model which approximates the real physical behavior. This is given by the so called *shape functions*, which interpolate the displacement at every nodal point.
- Setting of the (discretized) governing equations for the single element.
- Assembling of the global stiffness matrix for the entire body from the individual element stiffness matrices.
- Solution of the system of linear or non-linear equations, after application of load or displacement boundary conditions and initial conditions. In such a way, unknown nodal displacement values are obtained.
- Obtaining relevant information at nodal points, such as strains or stresses.

In particular, there is a crucial difference in the third and the fifth step depending on whether the system of equations is linear or non-linear.

2.2.1. Finite element discretized governing equations

A weak form of the governing equations is obtained using the *principle of virtual work*, so that a simplified system of equations is obtained. For the sake of brevity, the derivation is briefly outlined only for the UL approach. However, the discretized equations for the TL approach can be easily obtained using a transformation of coordinates.

Let us introduce the concept of *virtual displacements*. The space of virtual displacements $\mathcal{U}$ is defined as:

$$\delta\mathbf{v}(\mathbf{X}) \in \mathcal{U} \qquad \mathcal{U} = \left\{\delta \mathrm{v}_{\mathrm{i}} | \delta \mathrm{v}_{\mathrm{i}} \in \mathcal{C}^0(\mathbf{X}), \quad \delta \mathrm{v}_{\mathrm{i}} = 0 \quad \text{on} \quad \Gamma_{\mathrm{u}}\right\} \tag{2.2.1.1}$$

Such space contains all the kinematically admissible displacements $\delta\mathrm{v}_{\mathrm{i}}$. By definition a virtual displacement is vanishing on the boundary of the material domain Γ_{u}. The principle of virtual work is obtained by taking the product between the equation of conservation of linear momentum with the virtual displacements $\delta\mathbf{v}$ and integrating over the current configuration Ω:

$$\int_{\Omega} \delta\mathbf{v} \cdot \left(\rho\mathbf{b} + \nabla \cdot \boldsymbol{\sigma} - \rho\frac{\partial\dot{\mathbf{u}}}{\partial t}\right) d\Omega \tag{2.2.1.2}$$

Developing Equation 2.2.1.2 using Gauss' theorem, the traction continuity condition and traction boundary condition leads to the following:

$$\underbrace{\int_{\Omega} \boldsymbol{\epsilon}^{T}(\delta\mathbf{v})\boldsymbol{\sigma}\, d\Omega}_{\delta W_{int}} - \underbrace{\int_{\Omega} \delta\mathbf{v} \cdot \rho\mathbf{b}\, d\Omega - \int_{\Gamma_t} \delta\mathbf{v} \cdot \bar{\mathbf{t}}\, d\Gamma}_{\delta W_{ext}} + \underbrace{\int_{\Omega} \delta\mathbf{v} \cdot \rho\frac{\partial\dot{\mathbf{u}}}{\partial t}\, d\Omega}_{\delta W_{kin}} = 0 \tag{2.2.1.3}$$

Where $\boldsymbol{\epsilon}$ is the virtual strain tensor, which is a function of the virtual displacements. Equation 2.2.1.3 is the final form of the *principle of virtual work*. This one states that the sum of the virtual internal work δW_{int}, virtual external work δW_{ext} and virtual kinetic work δW_{kin} must be equal to zero. It is worth noting that there is a substantial difference between the principle of virtual work and the strong form of the governing equations. Equation 2.2.1.3 involves a *fictitious* work calculated on a set of forces, stresses and kinematically admissible displacements. Therefore, there is no need for the forces and stresses to actually occur in the physical system, as there is no need for the assumed displacements to be actual displacements. Furthermore, the smoothness requirement on the virtual displacement functions $\delta\mathbf{v}$ is relaxed, as they need only to be $\mathcal{C}^0$. However, as the weak form and the strong form of the governing equations are equivalent, a solution for the weak form is a solution for the strong form as well.

The FEM discretized equations are then derived by discretizing the domain Ω into n_e elements with sub-domains Ω_e, leading to the following approximation:

$$\Omega \approx \bigcup_{n_e} \Omega_e$$

The elements are topologically defined by their nodes as shown in Figure 2.5.

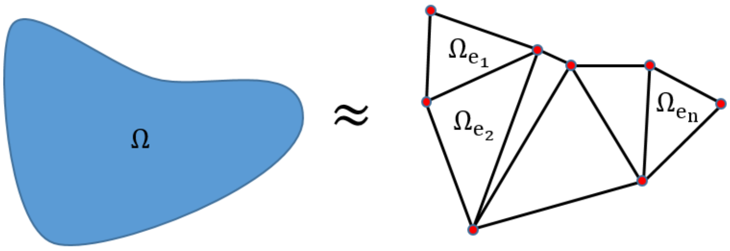


Figure 2.5: Visual example of FEM discretization

Let n_N be the total number of nodes in the domain. Denoting by $\mathbf{X}_I$ and $\mathbf{x}_I$, with $I = 1, ..., n_N$, the position vectors of the node I in the reference and current configuration respectively, the motion $\mathbf{x}(\mathbf{X}, t)$ in the FEM formulation is given by:

$$\mathbf{x}(\mathbf{X}, t) = \mathbf{x}_I(t) N_I(\mathbf{X}) \qquad \text{or} \qquad x_i(\mathbf{X}, t) = x_{iI}(t) N_I(\mathbf{X}) \tag{2.2.1.4}$$

Where $i = 1, ..., n_{dim}$ and $N_I(\mathbf{X})$ are the interpolating shape functions. Summation over the nodes has to be taken in Equation 2.2.1.4, even though the summation sign was here omitted for convenience.

In particular, it is required for the shape functions to satisfy the following condition:

$$\mathrm{N_I}(\mathbf{X}_\mathrm{J}) = \delta_\mathrm{IJ} \qquad \text{where} \qquad \begin{aligned} \delta_\mathrm{IJ} &= 1 \quad \text{if} \quad \mathrm{I} = \mathrm{J} \\ \delta_\mathrm{IJ} &= 0 \quad \text{if} \quad \mathrm{I} \neq \mathrm{J} \end{aligned} \tag{2.2.1.5}$$

The nodal displacement of the node I is given by the difference between current and reference position of the single node:

$$\mathbf{u}_\mathrm{I}(\mathrm{t}) = \mathbf{x}_\mathrm{I}(\mathrm{t}) - \mathbf{X}_\mathrm{I} \qquad \text{or} \qquad \mathrm{u_{iI}}(\mathrm{t}) = \mathrm{x_{iI}}(\mathrm{t}) - \mathrm{X_{iI}} \tag{2.2.1.6}$$

In a similar fashion, the displacement field can also be expressed in terms of the shape functions and the nodal displacements:

$$\begin{aligned} \mathbf{u}(\mathbf{X},\mathrm{t}) &= \mathbf{x}(\mathbf{X},\mathrm{t}) - \mathbf{X} \\ &= \mathbf{u}_\mathrm{I}(\mathrm{t})\mathrm{N_I}(\mathbf{X}) \end{aligned} \tag{2.2.1.7}$$

The velocity and acceleration fields are obtained by taking the material time derivative of the displacement:

$$\begin{aligned} \dot{\mathbf{u}}(\mathbf{X},\mathrm{t}) &= \frac{\partial \mathbf{u}(\mathbf{X},\mathrm{t})}{\partial \mathrm{t}} \qquad & \ddot{\mathbf{u}}(\mathbf{X},\mathrm{t}) &= \frac{\partial^2 \mathbf{u}(\mathbf{X},\mathrm{t})}{\partial \mathrm{t}^2} \\ &= \dot{\mathbf{u}}_\mathrm{I}(\mathrm{t})\mathrm{N_I}(\mathbf{X}) & &= \ddot{\mathbf{u}}_\mathrm{I}(\mathrm{t})\mathrm{N_I}(\mathbf{X}) \end{aligned} \tag{2.2.1.8}$$

Given the arbitrariness of the virtual displacement functions, expressing the latter in terms of FE shape functions, substituting in Equation 2.2.1.3 and writing the equations in a matrix form:

$$\int_\Omega \mathbf{B}^\mathrm{T}\boldsymbol{\sigma}\,\mathrm{d}\Omega - \int_\Omega \mathbf{N}^\mathrm{T}\rho\mathbf{b}\,\mathrm{d}\Omega - \int_{\Gamma_\mathrm{t}} \mathbf{N}^\mathrm{T}\bar{\mathbf{t}}\,\mathrm{d}\Gamma + \int_\Omega \mathbf{N}^\mathrm{T}\rho\ddot{\mathbf{u}}\,\mathrm{d}\Omega = 0 \tag{2.2.1.9}$$

Where $\mathbf{N}$ is the shape functions matrix and $\mathbf{B}$ is the shape functions derivative matrix, also called *strain-displacement matrix*, whose form depends on the finite element. Again, it is possible to physically identify each term of Equation 2.2.1.9:

$$\mathbf{f}^\mathrm{int} = \int_\Omega \mathbf{B}^\mathrm{T}\boldsymbol{\sigma}\,\mathrm{d}\Omega \tag{2.2.1.10}$$

$$\mathbf{f}^\mathrm{ext} = \int_\Omega \mathbf{N}^\mathrm{T}\rho\mathbf{b}\,\mathrm{d}\Omega + \int_{\Gamma_\mathrm{t}} \mathbf{N}^\mathrm{T}\bar{\mathbf{t}}\,\mathrm{d}\Gamma \tag{2.2.1.11}$$

$$\mathbf{f}^\mathrm{kin} = \int_\Omega \mathbf{N}^\mathrm{T}\rho\ddot{\mathbf{u}}\,\mathrm{d}\Omega \tag{2.2.1.12}$$

Where $\mathbf{f}^\mathrm{int}$, $\mathbf{f}^\mathrm{ext}$ and $\mathbf{f}^\mathrm{kin}$ are the internal, external and kinetic forces, respectively. The kinetic forces are usually defined as a product between the *mass matrix* and the nodal accelerations. Expressing the latter in terms of the shape functions and substituting in Equation 2.2.1.12 yields:

$$\mathbf{f}^\mathrm{kin} = \int_\Omega \mathbf{N}^\mathrm{T}\mathbf{N}\rho\ddot{\mathbf{u}}_\mathrm{J}\,\mathrm{d}\Omega \tag{2.2.1.13}$$

The mass matrix is then defined as:

$$\mathbf{M} = \int_{\Omega} \rho \mathbf{N}^{\mathrm{T}} \mathbf{N} \, \mathrm{d}\Omega \tag{2.2.1.14}$$

In a similar fashion, the internal forces $\mathbf{f}^{\mathrm{int}}$ can be expressed as a product between the element *stiffness matrix* $\mathbf{K}$ and the nodal displacement vector $\mathbf{u}$, with the stiffness matrix defined as such:

$$\mathbf{K} = \int_{\Omega} \mathbf{B}^{\mathrm{T}} \mathbf{D} \mathbf{B} \, \mathrm{d}\Omega \tag{2.2.1.15}$$

Where the Hooke's law relation between the stress tensor $\boldsymbol{\sigma}$ and the strain tensor $\boldsymbol{\epsilon}$ has been used, with $\mathbf{D}$ being material constitutive tensor. Therefore, the set of discretized FEM equations is given by:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}^{\mathrm{ext}} \qquad \rightarrow \qquad \mathbf{M}\ddot{\mathbf{u}} = \mathbf{f} \tag{2.2.1.16}$$

Where the arbitrariness of the virtual nodal displacements was used on the nodes not belonging to Γ_{u} and $\mathbf{f} = \mathbf{f}^{\mathrm{ext}} - \mathbf{f}^{\mathrm{int}}$. This consist in a system of $\mathrm{n}_{\mathrm{DOF}}$ equations in the nodal displacements, where $\mathrm{n}_{\mathrm{DOF}}$ is the number of nodal degrees of freedom not prescribed. For static problems, the inertial forces are equal to zero since the accelerations are zero and the governing equations are reduced to the equilibrium equations:

$$\mathbf{f}^{\mathrm{int}} = \mathbf{f}^{\mathrm{ext}} \tag{2.2.1.17}$$

It is worth mentioning that in FEM the system of equations is expressed in a *residual form* and solved with an iterative solver. On the other hand, the exact integration is replaced by numerical integration. Gaussian quadrature rule is most commonly used for this purpose. This integration technique, using an optimal choice of *quadrature points* and related weights, is able to give a very accurate approximation of the exact integral. The detailed procedure will be addressed more in detail in Section 2.3.

In the next section, the concept of *geometric non-linearity* will be introduced to the reader to acquire familiarity with the topic. At the end of the section, a popular geometric non-linear benchmark is reproduced to provide a deeper insight on the modelling of geometric non-linear problems in a FEM framework.

2.2.2. Geometric non-linearity in FEM

Considering a general structural problem, if a linear FEM analysis is performed, the response of the structure is assumed to be directly proportional to the applied load. The implicit assumptions of a linear analysis are the following:

- Displacements/rotations and strains are small (deformed configuration is assumed equal to the undeformed configuration throughout the analysis).
- Boundary conditions do not change during the analysis.
- The material laws do not change during the analysis (linear relation between stresses and strains).
- Applied loads are not changing directions throughout the analysis.

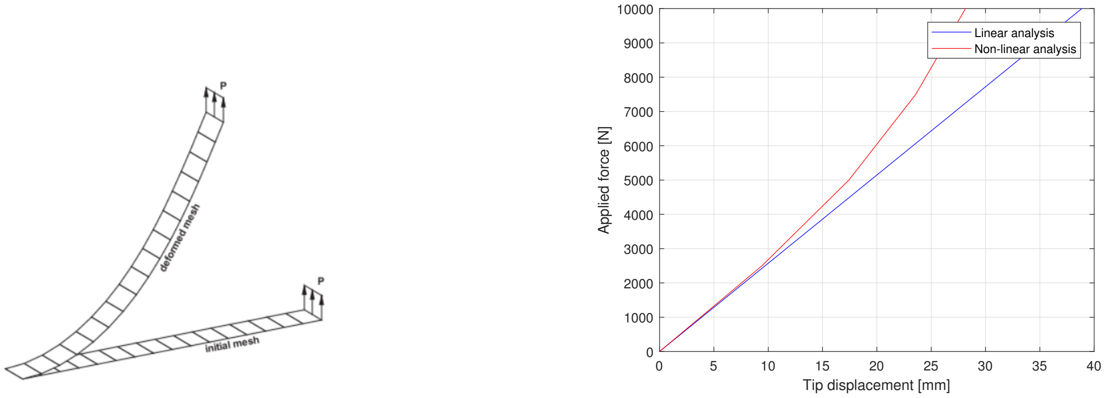
When geometric non-linear effects are considered, the first assumption does not hold anymore. In such a case, geometric changes in the structure are so large that the deformed configuration cannot be approximated to the undeformed one. As a result, if geometry changes are not considered, the load-deformation behavior is not well captured. Under such circumstances, there is a difference whether the problem variables are referred to the undeformed or to the deformed configuration [2, 52, 55]. To this regard, Bathe defined two types of geometric non-linearities [54]:

- Geometric non-linearities characterized by small strains but large displacements and rotations.
- Geometric non-linearities characterized by both large strains and large displacements and rotations.

The first class is of main interest for this project, as composite structures are known to experience small strains up to failure, even though they can undergo large displacements and rotations.

Cantilever beam with applied end shear load

In order to highlight relevant features regarding geometric non-linear FEM modelling, a popular benchmark problem proposed by many authors has been here reproduced using the FEM software Abaqus [17, 24, 56]. The model consists in an isotropic cantilever beam subjected to an end shear load, as illustrated in Figure 2.6a. A striking difference between the linear and non-linear case can be observed in the load-displacement equilibrium path, which is illustrated in Figure 2.6b.

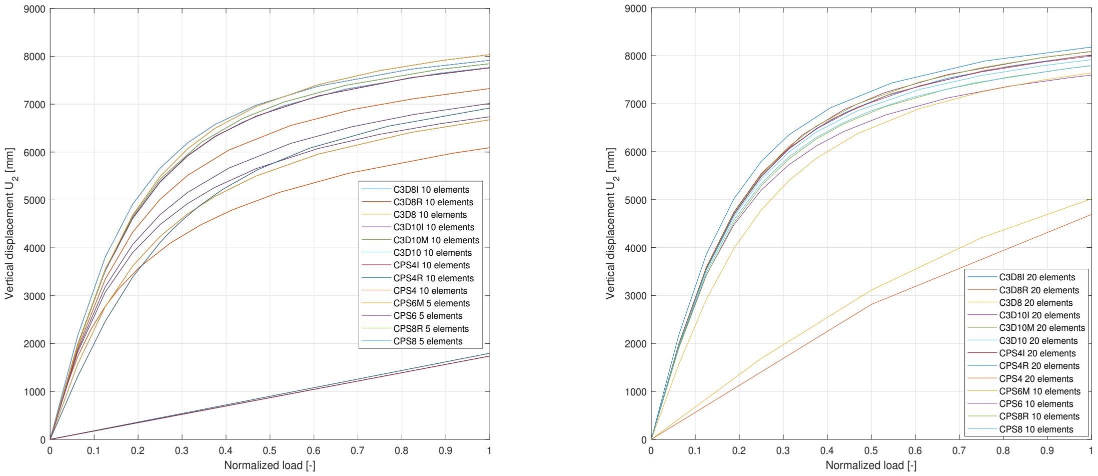


(a) Illustration of the benchmark model [56]

(b) Comparison between linear and geometric non-linear case

Figure 2.6: Cantilever beam subjected to end shear load

In particular, a stiffening behavior is observed in the geometric non-linear model. This is due to the fact that the geometry of the beam adapts to the applied load in order to withstand it more efficiently. Above a certain load level, an increase in load will cause a smaller and smaller increase in displacement at the tip of the beam. To provide an insight into the element performance, the same benchmark has been reproduced using different element types, as proposed in the Abaqus Benchmarks Guide [57]. The results obtained with a coarse and a fine mesh are shown in Figure 2.7a and 2.7b, respectively.



(a) Results for a coarse mesh

(b) Results for a fine mesh

Figure 2.7: Cantilever beam subjected to end shear load: element performance study

From the comparison it emerges that elements using a reduced integration scheme provide more accurate results for the displacement, in close agreement with the analytical solution by Bisshopp and Drucker [58]. On the other hand, elements using a full integration scheme, such as plane stress CPS4 and linear brick elements C3D8, are characterized by a very stiff behavior, due to a phenomenon called *shear locking*. Shear locking is a numerical phenomenon that affects first order full integration elements in bending problems. According to beam theory, the element should deform such that both top and bottom surfaces acquire a curved shape. On the other hand, in FE analyses, first order full integration elements are deforming as shown in Figure 2.8, as their edges are not able to bend. Considering the horizontal and vertical dotted lines on the surface of the element, these should ideally maintain an angle equal to 90 degrees. However, the top part is subjected to a compressive stress, while the bottom undergoes a tensile stress, resulting in a different angle. Therefore, a non-physical shear strain, that should be theoretically equal to zero, is introduced in the element. For this reason, it is also referred as *parasitic shear strain*. Such error arise from the fact that the displacement field is incorrectly approximated by a linear interpolation, due to the positions of the integration points using a full integration scheme. The magnitude of the error is observed to increase with increasing aspect ratio of the element. The parasitic shear is related to a fictitious strain energy component that physically does not exist. This increase in strain energy is responsible for the observed higher stiffness of the element, leading to an underestimation of the deflection [59].

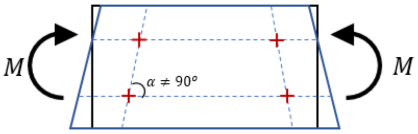


Figure 2.8: Shear locking in a linear full integration element

Many different strategies can be used in order to overcome the shear locking effect:

- *Reduced/Selective integration linear elements*: with these elements, the shear locking effect is eliminated, as the transverse shear terms are integrated at the centroid of the element.
- *Second-order elements*: these elements are able to represent quadratic displacement fields, due to their higher order interpolation. For this reason, they are more suitable for bending problems.
- *Incompatible mode elements*: these are first order fully integrated elements, which degrees of freedom are enhanced in order to better capture bending.

2.3. The 8-node linear hexahedral element

In this section, the implementation of the 8-node linear hexahedral FE is presented, which is the equivalent of the linear C3D8 element in Abaqus [59]. The topology of the element is illustrated in Figure 2.9. This FE is one of the most used in three-dimensional analyses.

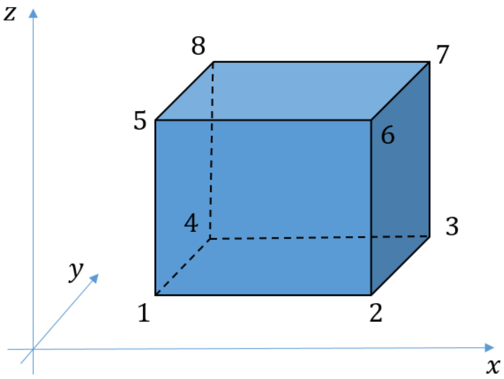


Figure 2.9: Topology of the 8-node hexahedral element

2.3.1. Geometric features and node numbering

The hexahedral element is characterized by the following geometric features:

- Eight corners, which are the physical nodes of the element.
- Six faces, each of them being defined by four nodes.
- Twelve edges, each of them defined by two nodes.

Considering a single element, every single node is given a number from 1 to 8 in ascending order. The numbering has to be such that the element volume is positive, which means the Jacobian determinant, defined by Equation 2.1.2.9, must be positive at every point. Therefore, the numbering convention, in agreement with the Analysis User's Abaqus Manual [59], is done according to the following steps:

- Choice of a starting node, which is given the number 1, and an initial face having this node as one of its corners.
- The other three nodes on the face are numbered counterclockwise (2, 3, 4) when this is looked at from the opposite face.
- The nodes on the face opposite to the initial one are given the numbers 5, 6, 7, 8, respectively.

2.3.2. FE formulation

The isoparametric, or *natural* coordinates of the element are indicated by the greek letters ξ, η, ζ. These ones vary from a value of -1 on one face to $+1$ on the opposite face. The specific values are provided in Table 2.3.

Node #	ξ	η	ζ	**Node** #	ξ	η	ζ
1	-1	-1	-1	5	-1	-1	$+1$
2	$+1$	-1	-1	6	$+1$	-1	$+1$
3	$+1$	$+1$	-1	7	$+1$	$+1$	$+1$
4	-1	$+1$	-1	8	-1	$+1$	$+1$

Table 2.3: Natural nodal coordinates for the hexahedral element

Given the physical coordinates (x_i, y_i, z_i) and the translational displacements of the nodes of the nodes (u_i, v_i, w_i), it is possible to express the position and the displacement in the element through the use of interpolating shape functions:

$$\begin{bmatrix} 1 \\ x \\ y \\ z \\ u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 \\ y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & y_7 & y_8 \\ z_1 & z_2 & z_3 & z_4 & z_5 & z_6 & z_7 & z_8 \\ u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 \\ v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 \\ w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7 & w_8 \end{bmatrix} \begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \\ N_6 \\ N_7 \\ N_8 \end{bmatrix} \tag{2.3.2.1}$$

Where $\mathrm{N_i}$ are the shape functions in terms of the physical coordinates. However, in the FE formulation, it is convenient to express the shape functions in terms of the element natural coordinates. The shape functions for the 8-node element are reported in Table 2.4.

$\mathrm{N_1} = \frac{1}{8}(1-\xi)(1-\eta)(1-\zeta)$	$\mathrm{N_5} = \frac{1}{8}(1-\xi)(1-\eta)(1+\zeta)$
$\mathrm{N_2} = \frac{1}{8}(1+\xi)(1-\eta)(1-\zeta)$	$\mathrm{N_6} = \frac{1}{8}(1+\xi)(1-\eta)(1+\zeta)$
$\mathrm{N_3} = \frac{1}{8}(1+\xi)(1+\eta)(1-\zeta)$	$\mathrm{N_7} = \frac{1}{8}(1+\xi)(1+\eta)(1+\zeta)$
$\mathrm{N_4} = \frac{1}{8}(1-\xi)(1+\eta)(1-\zeta)$	$\mathrm{N_8} = \frac{1}{8}(1-\xi)(1+\eta)(1+\zeta)$

Table 2.4: Shape functions of the hexahedral element

In order to obtain the derivatives of these with respect to the physical coordinates, the chain rule for partial differentiation has to be applied:

$$\begin{aligned}
\frac{\partial \mathrm{N_i}}{\partial \mathrm{x}} &= \frac{\partial \mathrm{N_i}}{\partial \xi}\frac{\partial \xi}{\partial \mathrm{x}} + \frac{\partial \mathrm{N_i}}{\partial \eta}\frac{\partial \eta}{\partial \mathrm{x}} + \frac{\partial \mathrm{N_i}}{\partial \zeta}\frac{\partial \zeta}{\partial \mathrm{x}}\\
\frac{\partial \mathrm{N_i}}{\partial \mathrm{y}} &= \frac{\partial \mathrm{N_i}}{\partial \xi}\frac{\partial \xi}{\partial \mathrm{y}} + \frac{\partial \mathrm{N_i}}{\partial \eta}\frac{\partial \eta}{\partial \mathrm{y}} + \frac{\partial \mathrm{N_i}}{\partial \zeta}\frac{\partial \zeta}{\partial \mathrm{y}}\\
\frac{\partial \mathrm{N_i}}{\partial \mathrm{z}} &= \frac{\partial \mathrm{N_i}}{\partial \xi}\frac{\partial \xi}{\partial \mathrm{z}} + \frac{\partial \mathrm{N_i}}{\partial \eta}\frac{\partial \eta}{\partial \mathrm{z}} + \frac{\partial \mathrm{N_i}}{\partial \zeta}\frac{\partial \zeta}{\partial \mathrm{z}}
\end{aligned} \tag{2.3.2.2}$$

These ones can also be expressed in matrix form as:

$$\begin{bmatrix} \frac{\partial \mathrm{N_i}}{\partial \mathrm{x}} \\ \frac{\partial \mathrm{N_i}}{\partial \mathrm{y}} \\ \frac{\partial \mathrm{N_i}}{\partial \mathrm{z}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \xi}{\partial \mathrm{x}} & \frac{\partial \eta}{\partial \mathrm{x}} & \frac{\partial \zeta}{\partial \mathrm{x}} \\ \frac{\partial \xi}{\partial \mathrm{y}} & \frac{\partial \eta}{\partial \mathrm{y}} & \frac{\partial \zeta}{\partial \mathrm{y}} \\ \frac{\partial \xi}{\partial \mathrm{z}} & \frac{\partial \eta}{\partial \mathrm{z}} & \frac{\partial \zeta}{\partial \mathrm{z}} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathrm{N_i}}{\partial \xi} \\ \frac{\partial \mathrm{N_i}}{\partial \eta} \\ \frac{\partial \mathrm{N_i}}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \mathbf{J} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial \mathrm{N_i}}{\partial \xi} \\ \frac{\partial \mathrm{N_i}}{\partial \eta} \\ \frac{\partial \mathrm{N_i}}{\partial \zeta} \end{bmatrix} \tag{2.3.2.3}$$

Where $\mathbf{J}$ is the Jacobian matrix, which transforms quantities from the cartesian coordinate system to the element natural coordinate system. On the other hand, the inverse of the Jacobian is relating derivatives with respect to the natural coordinates to derivatives with respect to the cartesian coordinates:

$$\mathbf{J} = \frac{\partial(\mathrm{x},\mathrm{y},\mathrm{z})}{\partial(\xi,\eta,\zeta)} \qquad \mathbf{J}^{-1} = \frac{\partial(\xi,\eta,\zeta)}{\partial(\mathrm{x},\mathrm{y},\mathrm{z})} \tag{2.3.2.4}$$

Recalling Equation 2.3.2.1, the Jacobian matrix is obtained by differentiating the relation between the cartesian and natural coordinates with respect to the natural coordinates. As a result, the Jacobian matrix is given by:

$$\mathbf{J} = \begin{bmatrix} \sum_\mathrm{i} \mathrm{x_i}\frac{\partial \mathrm{N_i}}{\partial \xi} & \sum_\mathrm{i} \mathrm{y_i}\frac{\partial \mathrm{N_i}}{\partial \xi} & \sum_\mathrm{i} \mathrm{z_i}\frac{\partial \mathrm{N_i}}{\partial \xi} \\ \sum_\mathrm{i} \mathrm{x_i}\frac{\partial \mathrm{N_i}}{\partial \eta} & \sum_\mathrm{i} \mathrm{y_i}\frac{\partial \mathrm{N_i}}{\partial \eta} & \sum_\mathrm{i} \mathrm{z_i}\frac{\partial \mathrm{N_i}}{\partial \eta} \\ \sum_\mathrm{i} \mathrm{x_i}\frac{\partial \mathrm{N_i}}{\partial \zeta} & \sum_\mathrm{i} \mathrm{y_i}\frac{\partial \mathrm{N_i}}{\partial \zeta} & \sum_\mathrm{i} \mathrm{z_i}\frac{\partial \mathrm{N_i}}{\partial \zeta} \end{bmatrix} \tag{2.3.2.5}$$

In which i indicates the summation over all the nodes. The shape function derivatives are needed to define the *strain-displacement matrix* $\mathbf{B}$. As a matter of fact, the strain in the element is obtained by deriving the displacement field, expressed in terms of the nodal displacements and shape functions in

Equation 2.3.2.1. Therefore, expressing the strain $\boldsymbol{\epsilon}$ in the element as a function of the $\mathbf{B}$ matrix and the nodal displacements:

$$\boldsymbol{\epsilon} = \mathbf{B}\mathbf{u} \qquad \rightarrow \qquad \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{zz} \\ 2\varepsilon_{xy} \\ 2\varepsilon_{yz} \\ 2\varepsilon_{zx} \end{bmatrix} = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & 0 & \dots & \frac{\partial N_8}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \dots & 0 & \frac{\partial N_8}{\partial y} & 0 \\ 0 & 0 & \frac{\partial N_1}{\partial z} & \dots & 0 & 0 & \frac{\partial N_8}{\partial z} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & 0 & \dots & \frac{\partial N_8}{\partial y} & \frac{\partial N_8}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial z} & \frac{\partial N_1}{\partial y} & \dots & 0 & \frac{\partial N_8}{\partial z} & \frac{\partial N_8}{\partial y} \\ \frac{\partial N_1}{\partial z} & 0 & \frac{\partial N_1}{\partial x} & \dots & \frac{\partial N_8}{\partial z} & 0 & \frac{\partial N_8}{\partial x} \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ w_1 \\ \vdots \\ u_8 \\ v_8 \\ w_8 \end{bmatrix} \tag{2.3.2.6}$$

The strain displacement matrix is used to build the stiffness matrix of the element. This one is obtained using Equation 2.2.1.15, once the constitutive relation matrix $\mathbf{D}$ is defined. Following a transformation from physical to parent element domain and applying Gaussian quadrature, the stiffness matrix is evaluated as follows:

$$\mathbf{K} = \sum_{k=1}^{n_Q} \mathbf{B}^T(\boldsymbol{\xi}_k)\mathbf{D}\mathbf{B}(\boldsymbol{\xi}_k)\det\mathbf{J}(\boldsymbol{\xi}_k)w(\boldsymbol{\xi}_k) \qquad \text{where} \qquad \boldsymbol{\xi}_k = (\xi_k, \eta_k, \zeta_k) \tag{2.3.2.7}$$

Where n_Q is the number of quadrature points and $\boldsymbol{\xi}_k$ are their natural coordinates in the element. The internal force vector $\mathbf{f}^{int}$, which is the element contribution to the residual force vector, is finally obtained by multiplication of the stiffness matrix $\mathbf{K}$ with the nodal displacements $\mathbf{u}$:

$$\mathbf{f}^{int} = \mathbf{K}\mathbf{u} \tag{2.3.2.8}$$

For the eight-node hexahedral element, the most used integration schemes in FEM are *full* and *reduced integration*. These ones are illustrated in Figure 2.10. The full integration element uses eight integration points. Their natural coordinates location and weights are reported in Table 2.5. On the other hand, the reduced integration uses only one quadrature point, located at the centroid of the element, with a weight equal to eight. The use of more integration points result in higher accuracy, even though this comes at the expenses of a higher computational time. Depending on the application, reduced integration might be preferred because it leads to considerable savings.

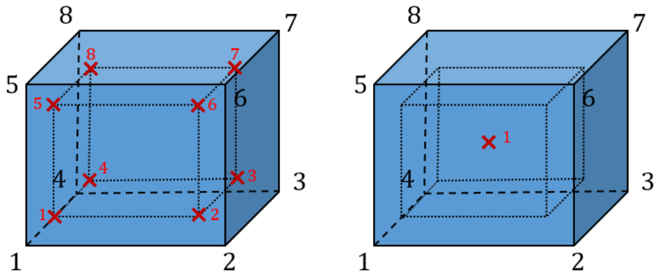


Figure 2.10: Integration points location in the hexahedral element for full (left) and reduced (right) integration schemes

Quad point #	(ξ,η,ζ)	w	**Quad point** #	(ξ,η,ζ)	w
1	$(-1/\sqrt{3},-1/\sqrt{3},-1/\sqrt{3})$	1	5	$(-1/\sqrt{3},-1/\sqrt{3},+1/\sqrt{3})$	1
2	$(+1/\sqrt{3},-1/\sqrt{3},-1/\sqrt{3})$	1	6	$(+1/\sqrt{3},-1/\sqrt{3},+1/\sqrt{3})$	1
3	$(-1/\sqrt{3},+1/\sqrt{3},-1/\sqrt{3})$	1	7	$(-1/\sqrt{3},+1/\sqrt{3},+1/\sqrt{3})$	1
4	$(+1/\sqrt{3},+1/\sqrt{3},-1/\sqrt{3})$	1	8	$(+1/\sqrt{3},+1/\sqrt{3},+1/\sqrt{3})$	1

Table 2.5: Quadrature points natural coordinates and weights for the full integration hexahedral element

2.3.3. Implementation of the linear brick element

As reported in the Abaqus User Subroutines Reference Guide, it is possible for users to implement several subroutines, which allow them to extend the software capabilities in order to meet certain analysis requirements [60]. User subroutines are compiled and linked to the main Abaqus executable prior the job execution. In this section, the implementation of the eight node linear hexahedral element, via an UEL subroutine in FORTRAN coding language, is presented. For the purpose of this project, the subroutine has been implemented for the Abaqus/Standard software interface. In the following sections, the implementation for static procedures only is presented. However, the implementation for dynamic procedures has also been investigated and can be found in Section A.1 in Appendix A. To facilitate the reader in understanding how user subroutines are interfacing with the software, the general structure of an Abaqus/Standard analysis is first presented. Next, the working scheme of the UEXTERNALDB subroutine, used as mean of communication between the UEL and Abaqus, is explained. Finally, a detailed description of the UEL subroutine is given.

Abaqus analysis flowchart

The UEL subroutine, which stands for UserELement subroutine, allows users to code finite elements in Abaqus. The UEL is used in conjunction with the UEXTERNALDB subroutine, which is needed to access user-defined external databases [60]. The general flowchart for the Abaqus/Standard procedure is given in Figure A.6 in Appendix A. The flowchart is showing how the UEXTERNALDB and the UEL subroutine are interfacing with the FE software.

UEXTERNALDB subroutine

The UEXTERNALDB subroutine is used as a means of communication between the UEL subroutine and Abaqus. This one is called four times during the whole process, the call being controlled through the use of the flag variable LOP:

- At the beginning of the analysis ($\mathrm{LOP} = 0$).
- At the beginning of each increment ($\mathrm{LOP} = 1$).
- At the end of each increment ($\mathrm{LOP} = 2$).
- At the end of the analysis ($\mathrm{LOP} = 3$).

By using this subroutine, it is possible to access purposely built external databases needed by the UEL subroutine as inputs for the analysis as well as to write the output results to external files. The working scheme of this subroutine is illustrated in Figure 2.11. At the beginning of the analysis, the subroutine is used to initialize relevant time parameters and load the external data-lists. When the increment starts, the analysis time parameters are updated. At the end of the increment, the output results are written to external files, following the VTK file formats [61]. At the end of the analysis, all the data-lists are cleaned to be ready for re-use.

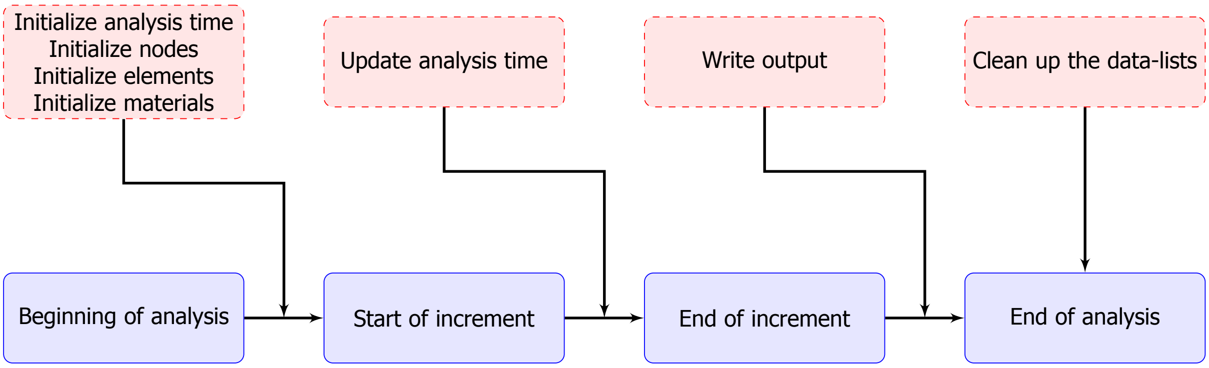


Figure 2.11: UEXTERNALDB subroutine flowchart

UEL subroutine

The UEL subroutine is used in Abaqus to define an user-element, linear or non-linear, with arbitrary complexity. Depending on the purpose of the element, the code must define the contribution of the element to the residual force vector, define the Jacobian matrix of the element, the mass matrix, the damping matrix and so on. The UEL subroutine interfaces with Abaqus each time element calculations have to be performed. Abaqus is providing the subroutine with interface parameters, among which the nodal coordinates and the nodal solutions (displacements, velocities, accelerations, etc.) for every DOF in the element. The full list of parameters is reported in the Abaqus User Subroutines Reference Manual [60]. In this section, the implementation of the element for static analyses is presented. Under such circumstances, the element's contribution to the Jacobian matrix and the element's contribution to the residual force vector must be coded by the user and returned to Abaqus through the parameters AMATRX and RHS, respectively. The flowchart of the implemented code is illustrated in Figure 2.12.

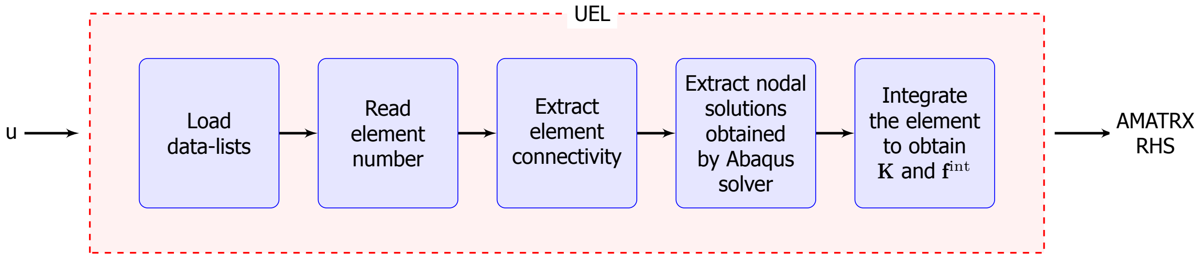


Figure 2.12: UEL subroutine flowchart

As previously mentioned, the UEXTERNALDB subroutine is used to access external databases. In particular, these contains an elements list, a nodes list and a materials list. The data-lists are loaded at the beginning of the UEL subroutine. Upon determination of the element number passed in by the Abaqus solver, the nodal connectivity of the element is read, which is used to extract the nodal coordinates and nodal solutions. This information is used to integrate the element and obtain its stiffness matrix $\mathbf{K}$ and internal force vector $\mathbf{f}^{\text{int}}$. Following the construction of the shape functions listed in Table 2.4, shape function derivatives and weights, the stiffness matrix is calculated with a loop over the integration points of the element:

- Obtain values of shape functions, shape functions derivatives and weights for the i-th integration point.
- Calculate jacobian of the element using Equation 2.3.2.5.
- Calculate the strain-displacement matrix using Equation 2.3.2.6.
- Calculate the strain in the element using the nodal solutions from Abaqus solver and the strain-displacement matrix.

- Calculate the stress in the element using the material constitutive relation.
- Calculate the contribution of the i-th integration point to the element stiffness matrix, using Equation 2.3.2.7.

Finally, the internal force vector of the element is obtained multiplying the stiffness matrix by the nodal displacements. The element's contribution to the Jacobian and to the residual force vector are then stored in the Abaqus parameters AMATRX and RHS respectively, and turned back to the Abaqus Solver. The system of equations is then solved by iterative Newton's method.

It is here worth to remind that multiple user elements can be coded in the same UEL subroutine and consequently used together. This particular feature come in handy for the FNM, and will be discussed more in detailed in Chapter 4. The user element can be invoked from the Abaqus input file through the following command:

```
*USER ELEMENT, TYPE=element_type, NODES=n

*ELEMENT, TYPE=element_type
```

Where an element type key has to be given in TYPE and n indicates the number of nodes in an element. Once an element type key is assigned, it uniquely relates to the user-defined element. Within an Abaqus UEL, the element can have an arbitrary number of nodes and degrees of freedom, provided that these are following the Abaqus convention [57].

2.3.4. User-element validation

In this section, the implemented element is validated through two benchmark problems. The first model consists in a single element uni-axial tensile test. The second, is the classical cantilever beam problem, which provides a multi-element model validation.

Uni-axial tension single element test

The reference model built in Abaqus is shown in Figure 2.13. The element is subjected to an uni-axial load in the positive y direction, while the following boundary conditions are applied:

- Nodes on the right face are constrained from moving in the x direction.
- Nodes on bottom left corner of the element are constrained from moving in the z direction.
- Nodes on the right corner of the right face are constrained from moving in the y direction.

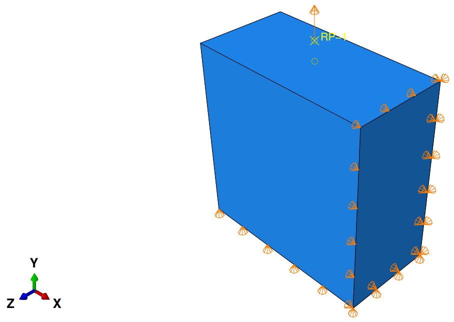
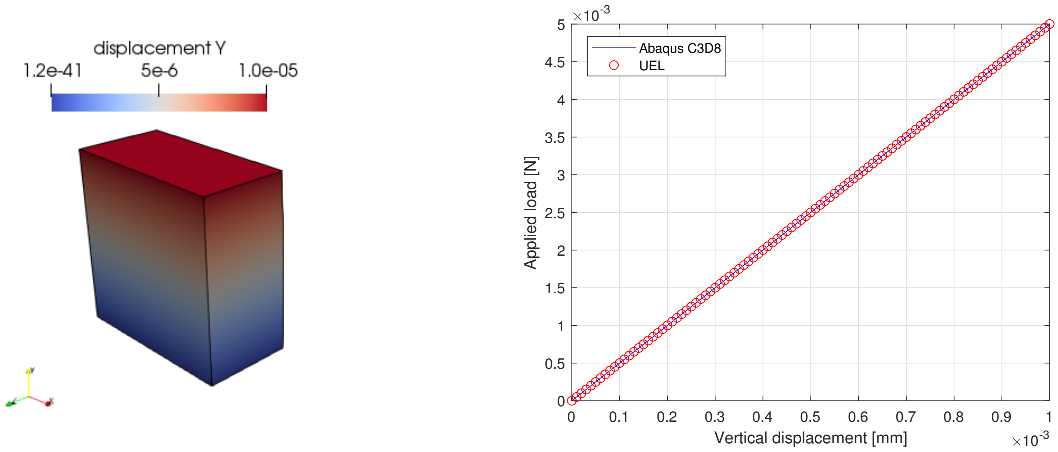


Figure 2.13: Reference model for the uni-axial single element test

The uni-axial load is applied through specification of a displacement load $\mathrm{U}_2 = 0.01\mathrm{mm}$ to the reference node *RP-1*, which is kinematically coupled to the top surface nodes of the element. The total load has been applied in 100 equal time increments. The dimensions of the element are $0.1\mathrm{mm}\,\mathrm{x}\,0.05\mathrm{mm}$. An isotropic material with elastic modulus $\mathrm{E} = 100\mathrm{MPa}$ and Poisson ratio $\nu = 0$ is used for this model.

Since the topology of the element is hidden inside the subroutine, it is not possible to visualize user elements with the standard post processing tool Abaqus/Viewer. For this reason, the third party open-source software ParaView from VTK has been used for data visualization. The contour plot of the vertical displacement for the user element is shown in Figure 2.14a. The comparison between the linear C3D8 Abaqus element and the implemented linear brick element is plotted in Figure 2.14b. As it can be observed, the two curves are superposing each other, showing the same structural response of the element.



(a) Vertical displacement contour plot

(b) Load-displacement curve comparison with Abaqus C3D8 linear element

Figure 2.14: Output results for the uni-axial single element model

Cantilever beam subjected to end shear load

In order to further validate the element, the cantilever beam model presented in Section 2.2.2 has been used as benchmark problem. The reference model built in Abaqus is reported in Figure 2.15. The beam has length $\ell = 10\mathrm{m}$, width $\mathrm{b} = 100\mathrm{mm}$ and height $\mathrm{h} = 147.8\mathrm{mm}$. An isotropic material with elastic modulus $\mathrm{E} = 100\mathrm{MPa}$ and Poisson ratio $\nu = 0$ is used for this model.

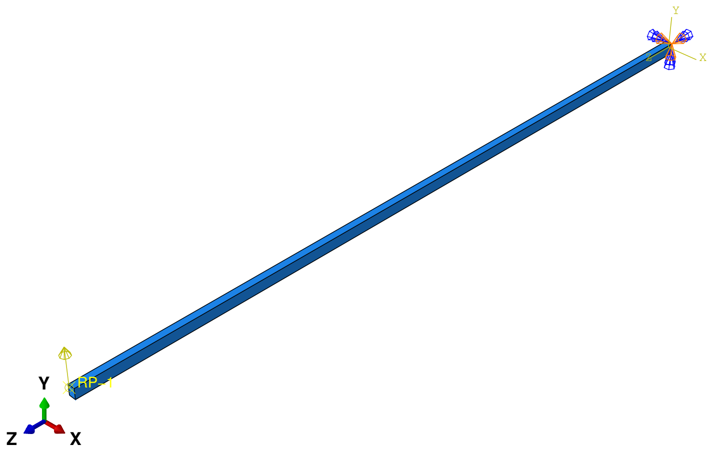
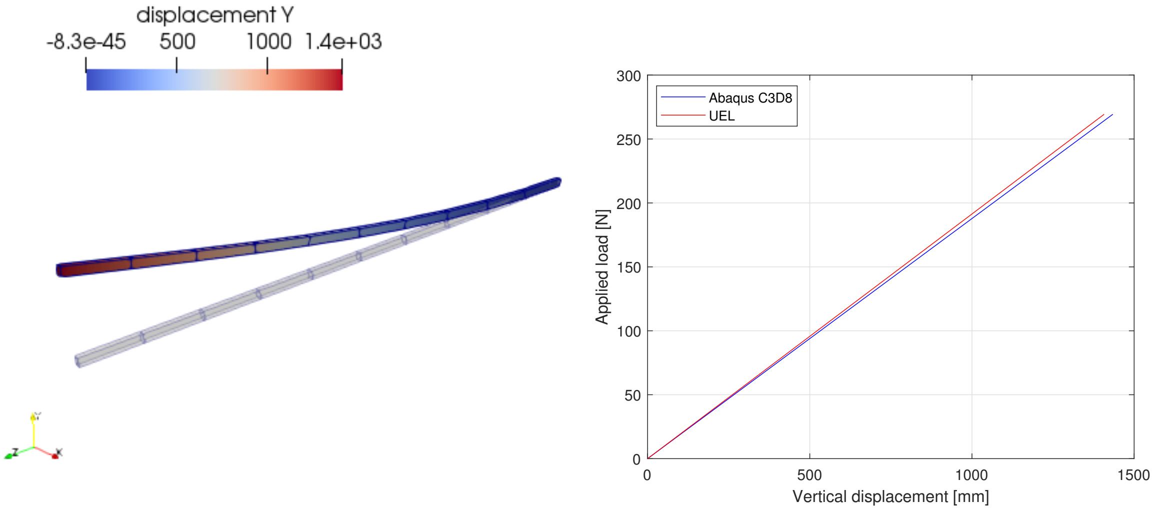


Figure 2.15: Reference model for the beam subjected to end shear load

The part has been meshed with 10 elements along the longitudinal direction of the beam and 1 element in the thickness direction of the beam. In the model, the right end has fixed boundary conditions, so that all the translations and rotations are constrained. A concentrated force $\mathrm{CF}_2 = 269.35\mathrm{N}$ is applied at the reference node *RP-1*, which is kinematically coupled to the left end. The material properties assigned to the part are the same as the ones used in the previous single element uni-axial tensile test. The deformed shape of the model is shown in Figure 3.11a, while the applied load as a function of the vertical displacement of the tip of the beam is plotted in Figure 2.16b, in comparison with the standard C3D8 Abaqus element.



(a) Vertical displacement contour plot

(b) Load-displacement curve comparison with Abaqus C3D8 linear element

Figure 2.16: Output results for the cantilever beam model

As it can be observed, the implemented brick element appears to have a slightly stiffer structural response. Nevertheless, the UEL can be considered to be in close agreement with the standard Abaqus element.

3

Co-rotational formulation

In this chapter, the CR formulation is presented in details. In the first part of this chapter, fundamental mathematical concepts on finite rotations are introduced to the reader. In a second instance, the CR approach is derived for a 3D solid brick element, starting from the involved kinematics to the derivation of the nodal force vector and tangent stiffness matrix used in the formulation. Finally, the formulation is applied to the hexahedral element implemented in Section 2.3 to extend its capabilities for geometric non-linear analysis. The co-rotational element is then validated reproducing three popular geometric non-linear benchmark problems.

3.1. Mathematics of finite rotations

In the 3D co-rotational approach, finite spatial rotations are considered. Hence, it is important to first introduce some relevant mathematical concepts, highlighting the difference between infinitesimal and finite rotations. The most important theorem for this subject is Euler's theorem [3]:

> *In three dimensions, the general displacement of a rigid body with one fixed point is a rotation about an axis that passes through that point*

While it is possible to assign a vectorial identity to infinitesimal rotations in space, this is not the case for finite rotations [15]. As a matter of fact, if we consider two or more finite rotations in space, these violate the commutative property of vectors, as changing their order leads to different results.

Let us consider the geometric construction illustrated in Figure 3.1. It will be shown that a finite rotation θ about an axis defined by the unit vector $\mathbf{e}$ consists in a one-to-one vector transformation, or a matrix. For the derivation it is useful to introduce the notion of *axial vector* or *pseudo-vector* $\boldsymbol{\theta}$:

$$\boldsymbol{\theta} = \theta\mathbf{e} \qquad\qquad \theta = (\phi^2 + \chi^2 + \psi^2)^{1/2} \tag{3.1.0.1}$$

Where ϕ, χ and ψ are its components in the cartesian system. The pseudo-vector is so called because it is a quantity that transforms as a vector but violates certain properties, such as the composition rule. In Figure 3.1, a point P with initial position vector $\mathbf{x}$ moves to occupy the position $\mathbf{x}^*$ indicated by the point P^*, so that:

$$\mathbf{x}^* = \mathbf{x} + \boldsymbol{\Delta}\mathbf{P} \tag{3.1.0.2}$$

The displacement of point P can also be written as:

$$\boldsymbol{\Delta}\mathrm{P} = \overline{\mathrm{PD}} + \overline{\mathrm{DP}^*} \tag{3.1.0.3}$$

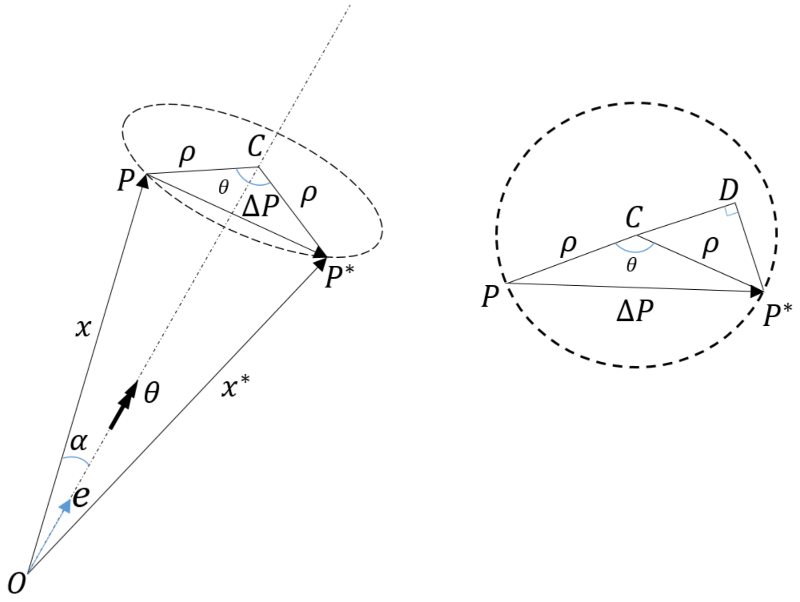


Figure 3.1: Finite rotation about an axis in space

Where $\overline{\mathbf{DP}^*}$ has direction $(\mathbf{e}\times\mathbf{x})$, so that it is normal to both $\overline{\mathbf{PC}}$ and $\mathbf{e}$. From Figure 3.1, it can be deduced that:

$$
\begin{aligned}
\overline{\mathbf{DP}^*} &= \rho \sin\theta && (3.1.0.4)\\
|\mathbf{e}\times\mathbf{x}| &= \rho && (3.1.0.5)
\end{aligned}
$$

Using these relations, the vector $\overline{\mathbf{DP}^*}$ can be written as:

$$
\begin{aligned}
\overline{\mathbf{DP}^*} &= \sin\theta(\mathbf{e}\times\mathbf{x})\\
&= \frac{\sin\theta}{\theta}(\boldsymbol{\theta}\times\mathbf{x}) && (3.1.0.6)
\end{aligned}
$$

On the other hand, it is clear from Figure 3.1 that the vector $\overline{\mathbf{PD}}$ is normal both to $\mathbf{e}$ and $\overline{\mathbf{DP}^*}$. As a result, its direction is given by $\mathbf{e}\times(\mathbf{e}\times\mathbf{x})$. Noting that the modulus of the latter is also ρ, the vector $\overline{\mathbf{PD}}$ can be expressed as:

$$
\begin{aligned}
\overline{\mathbf{PD}} &= (1-\cos\theta)(\mathbf{e}\times(\mathbf{e}\times\mathbf{x}))\\
&= 2\sin^2\frac{\theta}{2}(\mathbf{e}\times(\mathbf{e}\times\mathbf{x}))\\
&= \frac{1}{2}\frac{\sin^2(\theta/2)}{(\theta/2)^2}(\boldsymbol{\theta}\times(\boldsymbol{\theta}\times\mathbf{x})) && (3.1.0.7)
\end{aligned}
$$

Hence, the final position vector of P is given by the following:

$$
\mathbf{x}^{*} = \mathbf{x} + \frac{\sin\theta}{\theta}(\boldsymbol{\theta}\times\mathbf{x}) + \frac{1}{2}\frac{\sin^2(\theta/2)}{(\theta/2)^2}(\boldsymbol{\theta}\times(\boldsymbol{\theta}\times\mathbf{x})) \qquad (3.1.0.8)
$$

Introducing the auxiliary skew-symmetric matrix S containing the components of the pseudo-vector, often referred to with the name of *spinor*, it is possible to re-write the external products as follows:

$$\mathrm{S} = \begin{bmatrix} 0 & -\psi & \chi \\ \psi & 0 & \phi \\ -\chi & -\phi & 0 \end{bmatrix} \quad \rightarrow \quad \boldsymbol{\theta} \times \mathbf{x} = \mathrm{S}\mathbf{x} \quad \text{and} \quad \boldsymbol{\theta} \times (\boldsymbol{\theta} \times \mathbf{x}) = \mathrm{S}^2\mathbf{x} \tag{3.1.0.9}$$

Using the results derived so far, the final position vector of P can be expressed as:

$$\begin{aligned} \mathbf{x}^{*} &= \mathbf{x} + \frac{\sin\theta}{\theta}\mathrm{S}\mathbf{x} + \frac{1}{2}\frac{\sin^2(\theta/2)}{(\theta/2)^2}\mathbf{S}^{\mathbf{2}}\mathbf{x} \\ &= \mathbf{R}(\boldsymbol{\theta})\mathbf{x} \end{aligned} \tag{3.1.0.10}$$

Where we have established the *transformation rotation matrix* $\mathbf{R}$, which is a non-linear function of the pseudo-vector $\boldsymbol{\theta}$:

$$\mathbf{R}(\boldsymbol{\theta}) = \mathbf{I}_3 + \frac{\sin\theta}{\theta}\mathrm{S} + \frac{1}{2}\frac{\sin^2(\theta/2)}{(\theta/2)^2}\mathrm{S}^2 \tag{3.1.0.11}$$

This is also referred to in literature with the name of *rotator*. Such constructed matrix is *orthogonal*. This follows from the fact that S is an antisymmetrical matrix, so that:

$$\mathbf{R}^{\mathrm{T}} = \mathbf{I}_3 - \frac{\sin\theta}{\theta}\mathrm{S} + \frac{1}{2}\frac{\sin^2(\theta/2)}{(\theta/2)^2}\mathrm{S}^2 = \mathbf{R}(-\boldsymbol{\theta}) \tag{3.1.0.12}$$

Furthermore, let us consider a series of n consecutive rotations $\boldsymbol{\theta}_{\mathrm{i}}$, each of them characterized by their specific axis of rotation in space:

$$\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_{\mathrm{i}}, \ldots, \boldsymbol{\theta}_{\mathrm{n}-1}, \boldsymbol{\theta}_{\mathrm{n}}$$

It has been shown that it is possible to associate a transformation matrix to every individual rotation. Nevertheless, these sequence of rotations can be associated to a single transformation matrix $\mathbf{R}$ given by:

$$\mathbf{R} = \mathbf{R}_{\mathrm{n}}\mathbf{R}_{\mathrm{n}-1} \ldots \mathbf{R}_{\mathrm{i}} \ldots \mathbf{R}_2\mathbf{R}_1$$

On the other hand, if the inverse order of multiplication is considered:

$$\mathbf{R} = \mathbf{R}_1\mathbf{R}_2 \ldots \mathbf{R}_{\mathrm{i}} \ldots \mathbf{R}_{\mathrm{n}-1}\mathbf{R}_{\mathrm{n}}$$

The rotation matrix $\mathbf{R}$ will correspond to a sequence of rotations applied in the following order:

$$\boldsymbol{\theta}_{\mathrm{n}}, \boldsymbol{\theta}_{\mathrm{n}-1}, \ldots, \boldsymbol{\theta}_{\mathrm{i}}, \ldots, \boldsymbol{\theta}_2, \boldsymbol{\theta}_1$$

Hence, Equation 3.1.0.11 represents the linear one-to-one vector transformation that is associated to a finite rotation. It is worth noting that, if infinitesimal angles are considered ($\theta \to 0$), this one reduces to the standard equation for small rotations:

$$\mathbf{R} = \mathbf{I}_3 + \mathrm{S} \tag{3.1.0.13}$$

With Equation 3.1.0.11, it has been proved that it is possible to represent a finite rotation through a rotator which is a function of the pseudo-vector $\boldsymbol{\theta}$. Another possibility is to express the rotator in a series in S.

In order to do that, the trigonometric functions appearing in Equation 3.1.0.11 are expanded in a Taylor series in θ:

$$\begin{aligned}\mathbf{R} = \mathbf{I}_3 &+ \left(2 - \frac{\theta^3}{3!} + \frac{\theta^4}{5!} + \ldots + (-1)^{\mathrm{n}} \frac{\theta^{2\mathrm{n}}}{(2\mathrm{n}+1)!} \pm \ldots\right)\mathrm{S} \\ &+ \left(\frac{1}{2!} - \frac{\theta^2}{4!} + \frac{\theta^4}{6!} - \ldots + (-1)^{\mathrm{n}} \frac{\theta^{2\mathrm{n}}}{(2\mathrm{n}+2)!} \pm \ldots\right)\mathrm{S}^2 \end{aligned} \tag{3.1.0.14}$$

Next, observing that the powers of S can be calculated through the followings:

$$\begin{aligned}\mathrm{S}^{2\mathrm{n}} &= (-1)^{\mathrm{n}-1}\theta^{2(\mathrm{n}-1)}\mathrm{S}^2 \\ \mathrm{S}^{2\mathrm{n}-1} &= (-1)^{\mathrm{n}-1}\theta^{2(\mathrm{n}-1)}\mathrm{S}\end{aligned} \tag{3.1.0.15}$$

Combining Equation 3.1.0.14 and 3.1.0.15 the following expression of $\mathbf{R}$ is obtained:

$$\begin{aligned}\mathbf{R} &= \mathbf{I}_3 + \mathrm{S} + \frac{1}{2!}\mathrm{S}^2 + \frac{1}{3!}\mathrm{S}^3 + \ldots + \frac{1}{\mathrm{n}!}\mathrm{S}^{\mathrm{n}} + \ldots \\ &= \mathrm{e}^{\mathrm{S}}\end{aligned} \tag{3.1.0.16}$$

In literature, this one is often referred as *Rodrigues' formula* or *exponential map*. In this section, it has been shown that a finite rotation in space may be given different mathematical representations. The first one that has been introduced is the *axial vector* or *pseudo-vector*. Secondly, it has been proved that a finite rotation can also be described by an orthogonal matrix, which is given the name of *rotator*. Lastly, a finite rotation can be associated to a skew-symmetric matrix, referred to with the name of *spinor*. These ones are illustrated in Figure 3.2.

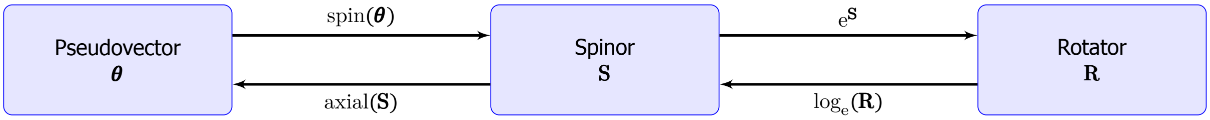


Figure 3.2: Relations between different representations of finite spatial rotations

In order to pass from a certain representation to another, specific operations exist, even though these ones will not be reported as they are not relevant to this work. It is worth mentioning that the rotator representation is the only one which is uniquely defined, as several normalizations can be used for the pseudo-vector and spinor [3]. An alternative parametrization often used in multi-body dynamics and robotics is the *quaternion parametrization* [62]. This one proved to be able to avoid singularities, making it numerically more stable, and can be obtained from the rotator representation through the Spurrier's quaternion extraction algorithm [63].

For the sake of this project, the rotator representation is relevant. This one will be used in the CR formulation to track the rigid body motion of the element. In particular, it will be shown that the rotator representation is intrinsically linked to the polar decomposition of the deformation gradient matrix.

3.2. Co-rotational approach for 3D solid hexahedral elements

In this section, previously introduced concepts are used to develop a CR procedure for 3D solid hexahedral elements in order to address geometric non-linear problems characterized by large displacements/rotations but small strains. The formulation outlined here follows the work by Moita and Crisfield [18, 21]. As previously anticipated, the CR approach, being an *element-independent* formulation, provides a relatively easy way to account for geometric non-linear effects retaining considerable flexibility [24]. The element-independent attribute refers to the fact that this technique allows re-use of already implemented linear elements and, once implemented for a certain element, it takes little effort to extend the formulation to elements belonging to the same class.

3.2.1. Co-rotational framework for geometric non-linear analyses

In this approach, an arbitrary motion is decomposed into two steps: one characterized by rigid body motion only and one by pure relative deformation. In a FEM framework, the decomposition is achieved through the definition of an embedded co-rotated frame for every element of the discretized domain. Such frame translates and rotates with the element but does not deform, so that orthogonality is maintained. The latter is fundamental in order to achieve an exact decomposition of rigid-body and deformational motions [3].

In order to obtain the orientation of the CR frame, the *polar decomposition theorem* is applied at the centroid of the element. Using the polar decomposition, the deformation gradient, which represents the arbitrary motion of the element, is decomposed into an orthogonal matrix representing the rigid body motion and a symmetric positive definite matrix, which represents the stretches in the element. In Section 2.1, it has been shown that this one can be performed using either the left stretch tensor $\mathbf{V}$ or the right stretch tensor $\mathbf{U}$ definition. In the first case, the element is first rotated and then strained, while in the second it is first strained and later rotated. It is clear from Figure 3.3, where the two-dimensional case is illustrated for simplicity, that there is no difference between the two as they lead to the same final configuration. Throughout this work, the right stretch tensor definition will be used.

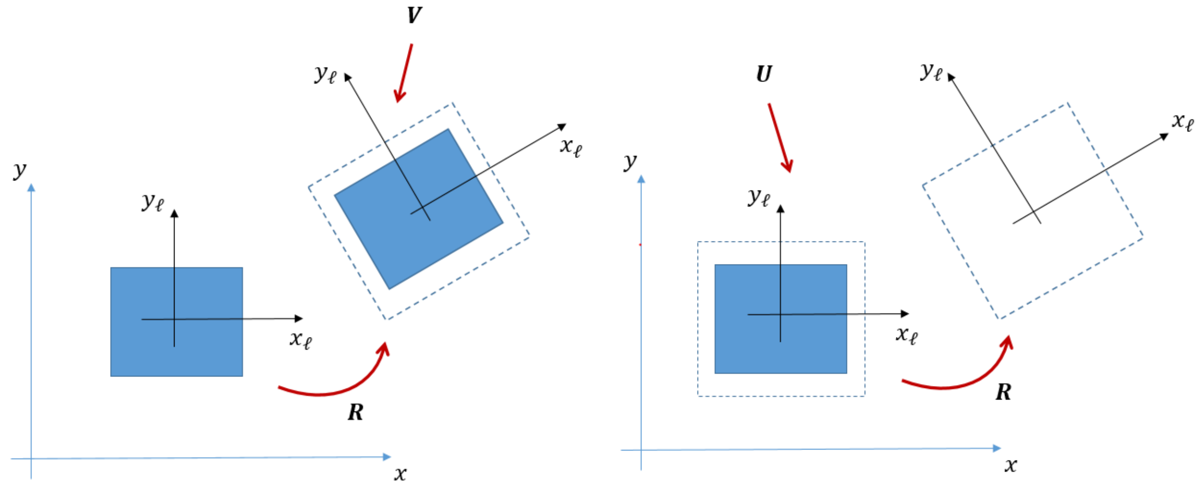


Figure 3.3: Polar decomposition of an arbitrary motion: left stretch tensor definition (on the left) and right stretch tensor definition (on the right)

Generally, in a geometric non-linear formulation the higher order terms of the strain tensor have to be taken into account. Nevertheless, by removal of rigid body displacements, if the strains are sufficiently small, linear assumptions are still valid in the CR reference frame. As a result, only the conventional linear strain-displacement matrix can be used in the formulation, neglecting the higher order terms.

In the derivation, three different configurations are considered: the *base* or *undeformed configuration*, the *co-rotated configuration* and the *deformed* or *final configuration*. It is worth to note that the CR configuration is a fictitious configuration, not really occupied by the element at any time during its motion. At the same time, two reference systems are considered: the *global reference system* and the *element local* or *CR reference system*. The latter is defined by the base of vectors $(\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3)$, and it is assumed to be aligned with the global reference system at the beginning of the analysis. By defining such local frame, it is possible come up with an expression for the local nodal displacements $\mathbf{u}_\ell$ in the form:

$$\mathbf{u}_\ell = f(\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3, \mathbf{u}_g) \tag{3.2.1.1}$$

Where $\mathbf{u}_g$ are the global nodal displacements. The such obtained local nodal displacements are then used to calculate the local stiffness matrix $\mathbf{K}_\ell$ and the local internal force vector $\mathbf{f}_\ell^{\mathrm{int}}$ using the followings:

$$\mathbf{K}_\ell = \int_{\Omega_\mathrm{e}} \mathbf{B}_\ell^\mathrm{T} \mathbf{D} \mathbf{B}_\ell \, \mathrm{d}\Omega_\mathrm{e} \tag{3.2.1.2}$$

$$\mathbf{f}_\ell^{\mathrm{int}} = \mathbf{K}_\ell \mathbf{u}_\ell \tag{3.2.1.3}$$

Where the subscript ℓ refers to quantities in the element local reference frame. Differentiating Equation 3.2.1.1, a transformation matrix $\mathbf{T}$ will be obtained, which relates the global nodal displacement variations to the local nodal displacement variations:

$$\delta \mathbf{u}_\ell = \mathbf{T} \delta \mathbf{u}_\mathrm{g} \tag{3.2.1.4}$$

From the equivalence of the principle of virtual work between the local and global frame, it follows that such obtained transformation matrix also relates the local internal force vector to the global internal force vector:

$$\mathbf{f}_\mathrm{g}^{\mathrm{int}} = \mathbf{T}^\mathrm{T} \mathbf{f}_\ell^{\mathrm{int}} \tag{3.2.1.5}$$

In order to obtain a consistent global tangent stiffness matrix, the differentiation of Equation 3.2.1.5 is needed. In a FEM framework, a consistent tangent stiffness means that the tangent stiffness is the gradient of the internal forces with respect to the global DOFs [3]. Even though an inconsistent tangent stiffness matrix does not change the equilibrium path, it is preferred to improve the convergence rate in an incremental-iterative solution. Therefore, differentiating Equation 3.2.1.5, the expression for the tangent stiffness matrix will be in the form:

$$\begin{aligned} \delta \mathbf{f}_\mathrm{g}^{\mathrm{int}} &= \mathbf{T}^\mathrm{T} \delta \mathbf{f}_\ell^{\mathrm{int}} + \delta \mathbf{T}^\mathrm{T} \mathbf{f}_\ell^{\mathrm{int}} \\ &= \mathbf{T}^\mathrm{T} \mathbf{K}_{\mathrm{t}\ell} \delta \mathbf{u}_\ell + \mathbf{K}_{\mathrm{tg}} \delta \mathbf{u}_\mathrm{g} \\ &= (\mathbf{T}^\mathrm{T} \mathbf{K}_{\mathrm{t}\ell} \mathbf{T} + \mathbf{K}_{\mathrm{tg}}) \delta \mathbf{u}_\mathrm{g} \end{aligned} \tag{3.2.1.6}$$

Where $\mathbf{K}_{\mathrm{t}\ell}$ is the local tangent stiffness matrix of the element, and $\mathbf{K}_{\mathrm{tg}}$ is the geometric stiffness contribution to the tangent stiffness matrix. Regarding the first contribution to the tangent stiffness, assuming a linear elastic constitutive law and small strains $\mathbf{K}_{\mathrm{t}\ell}$ is given by Equation 3.2.1.2. On the other hand, the geometric contribution to the tangent stiffness will be derived in the following section.

3.2.2. Derivation of transformation matrix and tangent stiffness matrix

Given the implementation of the linear hexahedral element defined in Section 2.3, the CR procedure outlined in this section serves to extend its validity to geometric non-linear analyses. Let us consider Figure 3.4. It is worth mentioning that such figure is illustrating the separation of rigid body motion using the left stretch tensor for visualization purposes, even though the right stretch tensor definition is used in the derivation.

As previously mentioned, two reference systems are considered: the global reference system, to which all the element calculations have to be referred and a local reference system which is embedded with the element. Let us denote with $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ the set of base vectors defining the global reference system, and with $(\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3)$ the set of base vectors defining the element local reference system. At the beginning of the analysis the element is occupying the configuration $\mathcal{C}_0$ and it is assumed that the element local axes are aligned with the global axes. As a result, the two set of base vectors are coincident. However, as the element undergoes motion, a procedure is needed to determine the orientation of the element local base vectors. Adopting a right stretch tensor definition, the deformation gradient $\mathbf{F}$ can be decomposed using the polar decomposition theorem as:

$$\mathbf{F} = \mathbf{R} \mathbf{U} \tag{3.2.2.1}$$

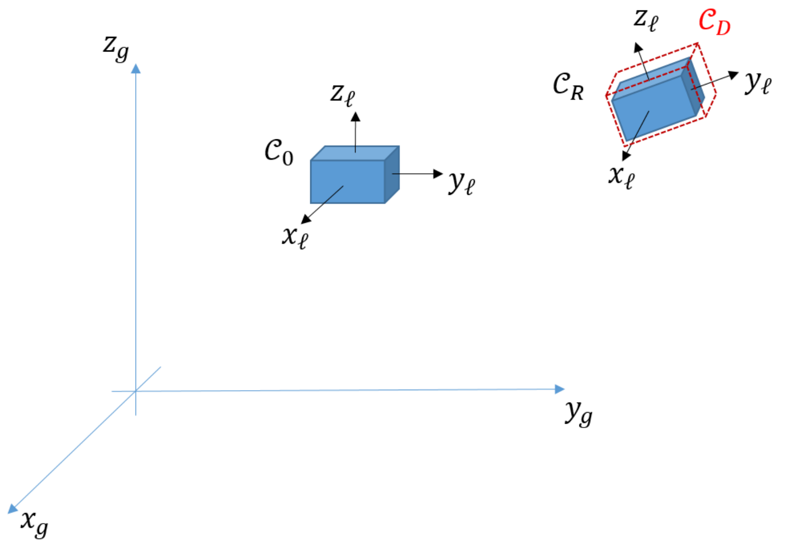


Figure 3.4: Visual illustration of initial ($\mathcal{C}_0$), co-rotated ($\mathcal{C}_\mathrm{R}$) and deformed ($\mathcal{C}_\mathrm{D}$) configurations

Where the deformation gradient, defined by Equation 2.1.2.6, is evaluated at the centroid of the element. The such obtained orthogonal matrix $\mathbf{R}$ is a rotator that transforms the base vectors $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ into $(\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3)$. The rotation matrix can also be written as follows:

$$\mathbf{R} = \begin{bmatrix} \hat{\mathrm{e}}_{1\mathrm{x}} & \hat{\mathrm{e}}_{2\mathrm{x}} & \hat{\mathrm{e}}_{3\mathrm{x}} \\ \hat{\mathrm{e}}_{1\mathrm{y}} & \hat{\mathrm{e}}_{2\mathrm{y}} & \hat{\mathrm{e}}_{3\mathrm{y}} \\ \hat{\mathrm{e}}_{1\mathrm{z}} & \hat{\mathrm{e}}_{2\mathrm{z}} & \hat{\mathrm{e}}_{3\mathrm{z}} \end{bmatrix} \tag{3.2.2.2}$$

Since the local axes are aligned with the global ones at the beginning of the analysis, if the local frame is located at the centroid of the element, the initial local coordinates for the node i are given by:

$$\mathbf{X}_\ell^\mathrm{i} = \mathbf{X}_\mathrm{g}^\mathrm{i} - \mathbf{X}_\mathrm{g}^\mathrm{c} \tag{3.2.2.3}$$

Where $\mathbf{X}_\mathrm{g}^\mathrm{c}$ refers to the initial global coordinates of the centroid of the element. On the other hand, using the rotation matrix $\mathbf{R}$, the deformed or final local coordinates for the node i are given by:

$$\begin{aligned} \mathbf{x}_\ell^\mathrm{i} &= \mathbf{X}_\ell^\mathrm{i} + \mathbf{u}_\ell^\mathrm{i} \\ &= \mathbf{R}^\mathrm{T}(\mathbf{x}_\mathrm{g}^\mathrm{i} - \mathbf{x}_\mathrm{g}^\mathrm{c}) \\ &= \mathbf{R}^\mathrm{T}\mathbf{x}_\mathrm{g}^\mathrm{ic} \end{aligned} \tag{3.2.2.4}$$

Where $\mathbf{x}_\mathrm{g}^\mathrm{c}$ are the deformed global coordinates of the centroid, and $\mathbf{x}_\mathrm{g}^\mathrm{ic}$ is used to indicate the difference between the deformed global coordinates of the i-th node and those of the centroid. From Equation 3.2.2.4, it is possible to deduce an expression for the local displacements of node i:

$$\mathbf{u}_\ell^\mathrm{i} = \mathbf{R}^\mathrm{T}\mathbf{x}_\mathrm{g}^\mathrm{ic} - \mathbf{X}_\ell^\mathrm{i} \tag{3.2.2.5}$$

These ones represent the pure deformational displacements in the element, expressed in the co-rotated or local reference frame. In order to obtain the transformation matrix which relates the variation of global to the variation of local displacements, Equation 3.2.2.5 has to be differentiated.

Hence, since the initial coordinates are constant, the following is obtained:

$$\begin{aligned}\delta\mathbf{u}_\ell^{\mathrm{i}} &= \delta\mathbf{R}^{\mathrm{T}}\mathbf{x}_{\mathrm{g}}^{\mathrm{ic}} + \mathbf{R}^{\mathrm{T}}\delta\mathbf{u}_{\mathrm{g}}^{\mathrm{ic}} \\ &= \delta\mathbf{R}^{\mathrm{T}}\mathbf{x}_{\mathrm{g}}^{\mathrm{ic}} + \mathbf{R}^{\mathrm{T}}\delta\mathbf{u}_{\mathrm{g}}^{\mathrm{i}}\end{aligned} \tag{3.2.2.6}$$

Where $\mathbf{R}^{\mathrm{T}}\delta\mathbf{u}_{\mathrm{g}}^{\mathrm{c}}$ has been added to the variation of local displacement vector in order to simplify the equations. Such modification does not change the results, if the zero strain condition in case of rigid body motion is satisfied [21]. Considering the variation of the rotation matrix $\mathbf{R}$, this one is obtained by differentiation of 3.1.0.16:

$$\delta\mathbf{R} = \mathrm{S}(\delta\boldsymbol{\theta})\mathbf{R} \tag{3.2.2.7}$$

Where $\mathrm{S}(\delta\boldsymbol{\theta})$ is the skew-symmetric matrix containing the components of the pseudo-vector $\delta\boldsymbol{\theta}$, which can be written in matrix form as:

$$\mathrm{S} = \begin{bmatrix} 0 & -\delta\theta_3 & \delta\theta_2 \\ \delta\theta_3 & 0 & -\delta\theta_1 \\ -\delta\theta_2 & \delta\theta_1 & 0 \end{bmatrix} \tag{3.2.2.8}$$

Hence, using the properties of transpose matrices together with the properties of skew-symmetric matrices the following is obtained:

$$\delta\mathbf{R}^{\mathrm{T}} = -\mathbf{R}^{\mathrm{T}}\mathrm{S}(\delta\boldsymbol{\theta}) \tag{3.2.2.9}$$

Substituting this result in 3.2.2.6:

$$\begin{aligned}\delta\mathbf{u}_\ell^{\mathrm{i}} &= -\mathbf{R}^{\mathrm{T}}\mathrm{S}(\delta\boldsymbol{\theta})\mathbf{x}_{\mathrm{g}}^{\mathrm{ic}} + \mathbf{R}^{\mathrm{T}}\delta\mathbf{u}_{\mathrm{g}}^{\mathrm{i}} \\ &= \mathbf{R}^{\mathrm{T}}\mathrm{S}(\mathbf{x}_{\mathrm{g}}^{\mathrm{ic}})\delta\boldsymbol{\theta} + \mathbf{R}^{\mathrm{T}}\delta\mathbf{u}_{\mathrm{g}}^{\mathrm{i}} \\ &= \mathbf{z}^{\mathrm{i}}\delta\boldsymbol{\theta} + \mathbf{R}^{\mathrm{T}}\delta\mathbf{u}_{\mathrm{g}}^{\mathrm{i}}\end{aligned} \tag{3.2.2.10}$$

Where the properties of cross-product between vectors has been used to obtain the last expression and where $\mathbf{z}^{\mathrm{i}}$ is defined as such:

$$\mathbf{z}^{\mathrm{i}} = \mathbf{R}^{\mathrm{T}}\begin{bmatrix} 0 & -(\mathrm{z}_{\mathrm{g}}^{\mathrm{i}} - \mathrm{z}_{\mathrm{g}}^{\mathrm{c}}) & (\mathrm{y}_{\mathrm{g}}^{\mathrm{i}} - \mathrm{y}_{\mathrm{g}}^{\mathrm{c}}) \\ (\mathrm{z}_{\mathrm{g}}^{\mathrm{i}} - \mathrm{z}_{\mathrm{g}}^{\mathrm{c}}) & 0 & -(\mathrm{x}_{\mathrm{g}}^{\mathrm{i}} - \mathrm{x}_{\mathrm{g}}^{\mathrm{c}}) \\ -(\mathrm{y}_{\mathrm{g}}^{\mathrm{i}} - \mathrm{y}_{\mathrm{g}}^{\mathrm{c}}) & (\mathrm{x}_{\mathrm{g}}^{\mathrm{i}} - \mathrm{x}_{\mathrm{g}}^{\mathrm{c}}) & 0 \end{bmatrix} \tag{3.2.2.11}$$

Writing Equation 3.2.2.10 for all the element nodes:

$$\begin{aligned}\delta\mathbf{u}_\ell &= \mathrm{col}(\mathbf{z}^{\mathrm{i}})\delta\boldsymbol{\theta} + \mathrm{diag}(\mathbf{R}^{\mathrm{T}})\delta\mathbf{u}_{\mathrm{g}} \\ &= \mathbf{z}\delta\boldsymbol{\theta} + \mathrm{diag}(\mathbf{R}^{\mathrm{T}})\delta\mathbf{u}_{\mathrm{g}}\end{aligned} \tag{3.2.2.12}$$

Where the $\mathrm{col}(\mathbf{z}^{\mathrm{i}})$ is a matrix with dimensions 24×3 formed by placing the matrices $\mathbf{z}^{\mathrm{i}}$ in a column matrix $\mathbf{z}$ and $\mathrm{diag}(\mathbf{R}^{\mathrm{T}})$ is a diagonal matrix formed by placing the matrix $\mathbf{R}^{\mathrm{T}}$ along its diagonal. At this point, it is necessary to express the variation $\delta\boldsymbol{\theta}$ as a function of the variation of global displacements $\delta\mathbf{u}_{\mathrm{g}}^{\mathrm{i}}$. In order to do that, the procedure given by Jetteur and Cescotto is followed [64]. Hence, the local spin $\boldsymbol{\Omega}_\ell$ evaluated at the centroid of the element is forced to be zero:

$$\boldsymbol{\Omega}_\ell = \mathbf{0} \tag{3.2.2.13}$$

This enforcement allows the element to pass the *large strain patch test* [21]. The local spin vector can also be written as:

$$\mathbf{\Omega}_\ell = \mathbf{A}_\ell^{\mathrm{T}} \mathbf{u}_\ell \tag{3.2.2.14}$$

Where $\mathbf{A}_\ell = \mathrm{row}(\mathbf{A}_\ell^{\mathrm{i}})$ is a row matrix with dimensions 3×24 formed using the matrices $\mathbf{A}_\ell^{\mathrm{i}}$, defined as follows:

$$\mathbf{A}_\ell^{\mathrm{i}} = \begin{bmatrix} 0 & -(\mathbf{J}_{31}^{-1}\mathrm{N}_\xi^{\mathrm{i}} + \mathbf{J}_{32}^{-1}\mathrm{N}_\eta^{\mathrm{i}} + \mathbf{J}_{33}^{-1}\mathrm{N}_\zeta^{\mathrm{i}}) & (\mathbf{J}_{21}^{-1}\mathrm{N}_\xi^{\mathrm{i}} + \mathbf{J}_{22}^{-1}\mathrm{N}_\eta^{\mathrm{i}} + \mathbf{J}_{23}^{-1}\mathrm{N}_\zeta^{\mathrm{i}}) \\ (\mathbf{J}_{31}^{-1}\mathrm{N}_\xi^{\mathrm{i}} + \mathbf{J}_{32}^{-1}\mathrm{N}_\eta^{\mathrm{i}} + \mathbf{J}_{33}^{-1}\mathrm{N}_\zeta^{\mathrm{i}}) & 0 & (\mathbf{J}_{11}^{-1}\mathrm{N}_\xi^{\mathrm{i}} + \mathbf{J}_{12}^{-1}\mathrm{N}_\eta^{\mathrm{i}} + \mathbf{J}_{13}^{-1}\mathrm{N}_\zeta^{\mathrm{i}}) \\ -(\mathbf{J}_{21}^{-1}\mathrm{N}_\xi^{\mathrm{i}} + \mathbf{J}_{22}^{-1}\mathrm{N}_\eta^{\mathrm{i}} + \mathbf{J}_{23}^{-1}\mathrm{N}_\zeta^{\mathrm{i}}) & -(\mathbf{J}_{11}^{-1}\mathrm{N}_\xi^{\mathrm{i}} + \mathbf{J}_{12}^{-1}\mathrm{N}_\eta^{\mathrm{i}} + \mathbf{J}_{13}^{-1}\mathrm{N}_\zeta^{\mathrm{i}}) & 0 \end{bmatrix} \tag{3.2.2.15}$$

Where $\mathbf{J}_{\mathrm{ij}}^{-1}$ represents the entry in the i-th row and j-th column of the inverse of the Jacobian matrix of the element evaluated at the centroid, while $\mathrm{N}_\xi^{\mathrm{i}}, \mathrm{N}_\eta^{\mathrm{i}}, \mathrm{N}_\zeta^{\mathrm{i}}$ are the shape function derivatives for the i-th node with respect to the natural coordinates. Therefore, differentiating Equation 3.2.2.14:

$$\begin{aligned} \delta\mathbf{\Omega}_\ell &= \mathbf{A}_\ell^{\mathrm{T}} \delta\mathbf{u}_\ell \\ &= \mathbf{A}_\ell^{\mathrm{T}} \mathrm{diag}(\mathbf{R}^{\mathrm{T}}) \delta\mathbf{u}_{\mathrm{g}} + \mathbf{A}_\ell^{\mathrm{T}} \mathbf{z} \delta\boldsymbol{\theta} \end{aligned} \tag{3.2.2.16}$$

From which it derives:

$$\begin{aligned} \delta\boldsymbol{\theta} &= -(\mathbf{A}_\ell^{\mathrm{T}} \mathbf{z})^{-1} \mathbf{A}_\ell^{\mathrm{T}} \mathrm{diag}(\mathbf{R}^{\mathrm{T}}) \delta\mathbf{u}_{\mathrm{g}} \\ &= \mathbf{v}^{\mathrm{T}} \delta\mathbf{u}_{\mathrm{g}} \end{aligned} \tag{3.2.2.17}$$

Where $\mathbf{v}$ is a 24×3 matrix. Using these results in Equation 3.2.2.12:

$$\begin{aligned} \delta\mathbf{u}_\ell &= (\mathrm{diag}(\mathbf{R}^{\mathrm{T}}) + \mathbf{z}\mathbf{v}^{\mathrm{T}}) \delta\mathbf{u}_{\mathrm{g}} \\ &= \mathbf{T} \delta\mathbf{u}_{\mathrm{g}} \end{aligned} \tag{3.2.2.18}$$

Therefore, $\mathbf{T}$ is the transformation matrix relating the global nodal displacement variations to the local nodal displacements variations. The next step is to derive a consistent global tangent stiffness. Recalling the equivalence of the principle of virtual work in the local and global frame, which implies Equation 3.2.1.5, and substituting the transformation matrix $\mathbf{T}$ in Equation 3.2.1.6, the second term becomes:

$$\begin{aligned} \mathbf{K}_{\mathrm{tg}} \delta\mathbf{u}_{\mathrm{g}} &= \delta\mathbf{T}^{\mathrm{T}} \mathbf{f}_\ell^{\mathrm{int}} \\ &= \delta(\mathrm{diag}(\mathbf{R}) + \mathbf{v}\mathbf{z}^{\mathrm{T}}) \mathbf{f}_\ell^{\mathrm{int}} \\ &= \underbrace{\delta\mathrm{diag}(\mathbf{R}) \mathbf{f}_\ell^{\mathrm{int}}}_{\text{term I}} + \underbrace{\delta\mathbf{v}\mathbf{z}^{\mathrm{T}} \mathbf{f}_\ell^{\mathrm{int}}}_{\text{term II}} \end{aligned} \tag{3.2.2.19}$$

Let us consider the two terms in Equation 3.2.2.19 separately. For the i-th node, term I can be written as:

$$\begin{aligned} \delta\mathbf{R}\mathbf{f}_\ell^{\mathrm{int,i}} &= \delta\mathbf{R}\mathbf{R}^{\mathrm{T}}\mathbf{R}\mathbf{f}_\ell^{\mathrm{int,i}} \\ &= \mathbf{S}(\delta\boldsymbol{\theta}) \bar{\mathbf{f}}_\ell^{\mathrm{int,i}} \end{aligned} \tag{3.2.2.20}$$

Where Equation 3.2.2.7 has been used and where $\bar{\mathbf{f}}_\ell^{\mathrm{int,i}} = \mathbf{R}\mathbf{f}_\ell^{\mathrm{int,i}}$.

Using the properties of cross-products and skew-symmetric matrices:

$$\begin{aligned}\delta\mathbf{R}\mathbf{f}_\ell^{\mathrm{int,i}} &= -\mathbf{S}(\bar{\mathbf{f}}_\ell^{\mathrm{int,i}})\delta\boldsymbol{\theta}\\ &= -\mathbf{S}(\bar{\mathbf{f}}_\ell^{\mathrm{int,i}})\mathbf{v}^{\mathrm{T}}\delta\mathbf{u}_{\mathrm{g}}\\ &= \mathbf{S}(\bar{\mathbf{f}}_\ell^{\mathrm{int,i}})^{\mathrm{T}}\mathbf{v}^{\mathrm{T}}\delta\mathbf{u}_{\mathrm{g}}\end{aligned} \tag{3.2.2.21}$$

On the other hand, term II in Equation 3.2.2.19 is given by:

$$\delta(\mathbf{v}\mathbf{z}^{\mathrm{T}})\mathbf{f}_\ell^{\mathrm{int,i}} = \underbrace{\mathbf{v}\delta\mathbf{z}^{\mathrm{T}}\mathbf{f}_\ell^{\mathrm{int,i}}}_{\text{term III}} + \underbrace{\delta\mathbf{v}\mathbf{z}^{\mathrm{T}}\mathbf{f}_\ell^{\mathrm{int,i}}}_{\text{term IV}} \tag{3.2.2.22}$$

Considering term III in Equation 3.2.2.22:

$$\begin{aligned}\mathbf{v}\delta\mathbf{z}^{\mathrm{T}}\mathbf{f}_\ell^{\mathrm{int,i}} &= \mathbf{v}\delta\sum_{\mathrm{i=1}}^{\mathrm{n_N}}\Big(\mathbf{R}^{\mathrm{T}}\mathbf{S}(\mathbf{x}_{\mathrm{g}}^{\mathrm{ic}})\Big)^{\mathrm{T}}\mathbf{f}_\ell^{\mathrm{int,i}}\\ &= -\mathbf{v}\delta\sum_{\mathrm{i=1}}^{\mathrm{n_N}}\Big(\mathbf{S}(\mathbf{x}_{\mathrm{g}}^{\mathrm{ic}})\mathbf{R}\Big)\mathbf{f}_\ell^{\mathrm{int,i}}\\ &= \underbrace{-\mathbf{v}\sum_{\mathrm{i=1}}^{\mathrm{n_N}}\delta\mathbf{S}(\mathbf{x}_{\mathrm{g}}^{\mathrm{ic}})\mathbf{R}\mathbf{f}_\ell^{\mathrm{int,i}}}_{\text{term V}}\underbrace{-\mathbf{v}\sum_{\mathrm{i=1}}^{\mathrm{n_N}}\mathbf{S}(\mathbf{x}_{\mathrm{g}}^{\mathrm{ic}})\delta\mathbf{R}\mathbf{f}_\ell^{\mathrm{int,i}}}_{\text{term VI}}\end{aligned} \tag{3.2.2.23}$$

Elaborating on term V of Equation 3.2.2.23:

$$\begin{aligned}\mathbf{v}\sum_{\mathrm{i=1}}^{\mathrm{n_N}}\Big(-\delta\mathbf{S}(\mathbf{x}_{\mathrm{g}}^{\mathrm{ic}})\Big)\mathbf{R}\mathbf{f}_\ell^{\mathrm{int,i}} &= -\mathbf{v}\sum_{\mathrm{i=1}}^{\mathrm{n_N}}\mathbf{S}(\delta\mathbf{u}_{\mathrm{g}})\mathbf{R}\mathbf{f}_\ell^{\mathrm{int,i}}\\ &= \mathbf{v}\sum_{\mathrm{i=1}}^{\mathrm{n_N}}\mathbf{S}(\bar{\mathbf{f}}_\ell^{\mathrm{int,i}})\delta\mathbf{u}_{\mathrm{g}}\end{aligned} \tag{3.2.2.24}$$

On the other hand, expanding term VI making use of Equation 3.2.2.7:

$$\begin{aligned}\mathbf{v}\sum_{\mathrm{i=1}}^{\mathrm{n_N}}\Big(-\mathbf{S}(\mathbf{x}_{\mathrm{g}}^{\mathrm{ic}})\delta\mathbf{R}\Big)\mathbf{f}_\ell^{\mathrm{int,i}} &= -\mathbf{v}\sum_{\mathrm{i=1}}^{\mathrm{n_N}}\mathbf{S}(\mathbf{x}_{\mathrm{g}}^{\mathrm{ic}})\mathbf{S}(\delta\boldsymbol{\theta})\mathbf{R}\mathbf{f}_\ell^{\mathrm{int,i}}\\ &= \mathbf{v}\sum_{\mathrm{i=1}}^{\mathrm{n_N}}\mathbf{S}(\mathbf{x}_{\mathrm{g}}^{\mathrm{ic}})\mathbf{S}(\bar{\mathbf{f}}_\ell^{\mathrm{int,i}})\delta\boldsymbol{\theta}\\ &= \mathbf{v}\sum_{\mathrm{i=1}}^{\mathrm{n_N}}\mathbf{S}(\mathbf{x}_{\mathrm{g}}^{\mathrm{ic}})\mathbf{S}(\bar{\mathbf{f}}_\ell^{\mathrm{int,i}})\mathbf{v}^{\mathrm{T}}\delta\mathbf{u}_{\mathrm{g}}\end{aligned} \tag{3.2.2.25}$$

It is worth noting that the expression obtained for term VI in Equation 3.2.2.25 is a non-symmetric term. However, it can be proven that the anti-symmetric part of Equation 3.2.2.25 vanishes at equilibrium. In order to do that, it is sufficient to observe that the anti-symmetric part of the central block of Equation 3.2.2.25 is given by:

$$\mathrm{asym}\Big(\mathbf{S}(\mathbf{x}_{\mathrm{g}}^{\mathrm{ic}})\mathbf{S}(\bar{\mathbf{f}}_\ell^{\mathrm{int,i}})\Big) = \frac{1}{2}\sum_{\mathrm{i=1}}^{\mathrm{n_N}}\Big(\mathbf{x}_{\mathrm{g}}^{\mathrm{ic}}{\bar{\mathbf{f}}_\ell^{\mathrm{int,i}}}^{\mathrm{T}} - {\mathbf{x}_{\mathrm{g}}^{\mathrm{ic}}}^{\mathrm{T}}\bar{\mathbf{f}}_\ell^{\mathrm{int,i}}\Big) \tag{3.2.2.26}$$

Equation 3.2.2.26 represents the three rotational equilibrium equations for the element and it is equal to zero at equilibrium. The last term that should be addressed is term IV in Equation 3.2.2.22. However, it can be observed that this is also including the rotational equilibrium equations, which at equilibrium are zero. Hence, at equilibrium, the element tangent stiffness matrix becomes symmetric, resulting in quadratic convergence for the incremental iterative Newton-Raphson algorithm [3, 26]. Furthermore, it has been proved by Nour-Omid and Rankin that, even if the anti-symmetric terms are neglected in the element tangent stiffness formulation, there is no deterioration of the quadratic rate of convergence [26].

As a result, combining Equation 3.2.2.21, 3.2.2.24 and 3.2.2.26, the geometric stiffness contribution to the element tangent stiffness is given by:

$$\mathbf{K}_{\mathrm{tg}} = -\mathrm{col}\Big(\mathbf{S}(\bar{\mathbf{f}}_{\ell}^{\mathrm{int,i}})\Big)\mathbf{v}^{\mathrm{T}} + \mathbf{v}\mathrm{row}\Big(\mathbf{S}(\bar{\mathbf{f}}_{\ell}^{\mathrm{int,i}})\Big) + \mathbf{v}\mathrm{sym}\Big(\sum_{\mathrm{i=1}}^{\mathrm{n_N}} \mathbf{S}(\mathbf{x}_{\mathrm{g}}^{\mathrm{ic}})\mathbf{S}(\bar{\mathbf{f}}_{\ell}^{\mathrm{int,i}})\Big)\mathbf{v}^{\mathrm{T}} \tag{3.2.2.27}$$

3.2.3. Implementation of the co-rotational technique

In the previous section, the transformation matrix and the tangent stiffness matrix were derived using a variational approach. Furthermore, it has been described how the CR approach can be applied to the hexahedral element implementation presented in Section 2.3. In this section, a more detailed overview is given regarding the implementation of the co-rotational approach.

The CR approach can be thought as a filtering operation, working by extraction and addition of rigid body motion in the element. In order to extract the rigid body displacements, it is necessary to obtain the rotation matrix $\mathbf{R}$, transforming quantities from global to co-rotated reference system. Such rotation matrix must be *orthonormal*, i.e. it must satisfy the followings:

- $\mathbf{R}^{\mathrm{T}}\mathbf{R} = \mathbf{I}$
- $\det(\mathbf{R}) = +1$

The first property translates into the orthogonality of the desired matrix, as every column is a unit length vector perpendicular to the others. On the other hand, the second property implies that the third column is given by the cross product between the first two. Orthogonality alone is not sufficient as the determinant of an orthogonal matrix can be either +1 or -1. The second case will result in a *reflection* in the rotation matrix, leading to an incorrect determination of the rigid body rotation.

As anticipated in the previous section, the desired rotation matrix is equivalent to the orthogonal polar factor of the the deformation gradient matrix, evaluated at the centroid of the element. Therefore, from a computational point of view, this translates into an *eigenvalue problem*. Indeed, starting from Equation 3.2.2.1, it is clear that:

$$\mathbf{F}^{\mathrm{T}}\mathbf{F} = \mathbf{U}^{\mathrm{T}}\mathbf{U} \tag{3.2.3.1}$$

Furthermore, it is known that the eigenvalues of a real symmetric matrix are real and they can be used with the eigenvectors to diagonalize such matrix. Denoting with $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ the eigenvectors of $\mathbf{U}$ and with $\lambda_1, \lambda_2, \lambda_3$ the associated eigenvalues, the followings are true:

$$\mathbf{U} = \mathbf{Q}^{\mathrm{T}}\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}\mathbf{Q} \qquad \mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{Q}^{\mathrm{T}}\begin{bmatrix} \lambda_1^2 & 0 & 0 \\ 0 & \lambda_2^2 & 0 \\ 0 & 0 & \lambda_3^2 \end{bmatrix}\mathbf{Q} \tag{3.2.3.2}$$

Where $\mathbf{Q}$ is the matrix whose columns are the eigenvectors of $\mathbf{U}$. Equation 3.2.3.2 is also referred to as *spectral decomposition of a matrix*. As a result, it is possible to obtain the matrix $\mathbf{U}$ by calculating the eigenvalues and eigenvectors of $\mathbf{U}^{\mathrm{T}}\mathbf{U}$ and taking the square root of the diagonal eigenvalues matrix. Next, the desired rotation matrix is given by:

$$\mathbf{R} = \mathbf{F}\mathbf{U}^{-1} \tag{3.2.3.3}$$

However, in the present implementation, the existing relation between polar decomposition and Singular Value Decomposition (SVD) has been used to obtain the rotation matrix. Using the SVD, the deformation gradient $\mathbf{F}$ is factorized into the product of three matrices [65]:

$$\mathbf{F} = \mathbf{W}\boldsymbol{\Sigma}\mathbf{P}^{\mathrm{T}} \tag{3.2.3.4}$$

Where $\mathbf{W}$ and $\mathbf{P}$ are orthogonal matrices, while $\boldsymbol{\Sigma}$ is a diagonal matrix with positive real entries. In particular, the decomposition is characterized by the following:

- $\mathbf{W}$ is the matrix containing the eigenvectors of $\mathbf{F}\mathbf{F}^{\mathrm{T}}$.
- $\mathbf{P}$ is the matrix containing the eigenvectors of $\mathbf{F}^{\mathrm{T}}\mathbf{F}$.
- $\boldsymbol{\Sigma}$ is a diagonal matrix containing the square root of the eigenvalues associated with the eigenvectors of $\mathbf{W}$ and $\mathbf{P}$, also referred to as *singular values*.

Starting from Equation 3.2.2.1, let us consider the right stretch tensor $\mathbf{U}$. Since this one is symmetric positive definite, the SVD of $\mathbf{U}$ is equal to its spectral decomposition. Denoting with $\boldsymbol{\Lambda}$ the diagonal matrix containing the eigenvalues of $\mathbf{U}$, it is clear that $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}$ and $\mathbf{P} = \mathbf{Q}^{\mathrm{T}}$. Therefore, using these results and combining Equation 3.2.2.1 with 3.2.3.4:

$$\mathbf{R}\mathbf{Q}^{\mathrm{T}}\boldsymbol{\Lambda}\mathbf{Q} = \mathbf{W}\boldsymbol{\Sigma}\mathbf{P}^{\mathrm{T}} \qquad \rightarrow \qquad \mathbf{R} = \mathbf{W}\mathbf{P}^{\mathrm{T}} \tag{3.2.3.5}$$

In order to avoid the above mentioned reflection case, the algorithm to extract the rotation matrix $\mathbf{R}$ has been modified according to the following:

$$\mathbf{R} = \mathbf{W}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{W}\mathbf{P}^{\mathrm{T}}) \end{bmatrix}\mathbf{P}^{\mathrm{T}} \tag{3.2.3.6}$$

Such modification ensures the orthonormality of the rotation matrix. Finally, once the rotation matrix $\mathbf{R}$ is obtained, the procedure to embed the hexahedral element in the co-rotational approach is illustrated in Figure A.7 in Appendix A.

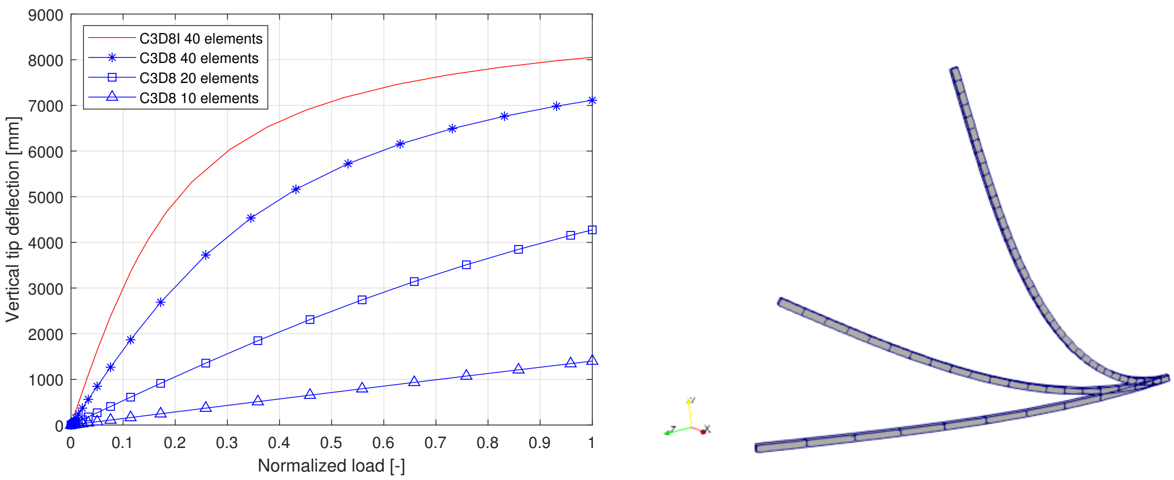
It is worth mentioning that the formulation presented in Section 3.2.2 has been successfully validated with only the first two benchmark problems that are proposed in Section 3.3. In particular, the inaccuracy of the solution for the third case is believed to be associated with a flaw in the implementation of the transformation matrix $\mathbf{T}$. As a result, a modified version of the formulation presented in Section 3.2.2 has been retained, which has been successfully validated with all the three benchmark problems proposed in Section 3.3. In the present implementation, the transformation matrix is given by the rotation matrix only and the tangent stiffness matrix is formed only with the first contribution appearing in Equation 3.2.1.6. Hence, the present implementation does not result in a fully consistent tangent stiffness matrix. An inconsistent tangent stiffness matrix is known to affect only the convergence rate of the solution, while not altering the accuracy of the solution [3]. However, in order to reach the full potential of the method, all the contributions to the tangent stiffness matrix should be included.

3.3. Validation of the non-linear 3D hexahedral element

In order to validate the performance of the element, three popular geometric non-linear benchmarks found in literature have been reproduced [56]. The obtained results are compared to the Abaqus standard non-linear elements, which are using an UL formulation [59]. In some of the models, selective integration has been used to eliminate the shear locking effect, typical of full integration schemes.

3.3.1. Beam subjected to end shear load

This validation model is the same as the one presented in Section 2.3.4. One thing to keep in mind when comparing the UEL results to the Abaqus C3D8 standard element is the influence of the shear locking effect, already described in Section 2.2.2. In the linear benchmark, the part was meshed using 1 element through the thickness and 10 along the longitudinal direction for simplicity. However, in the geometric non-linear case, mesh refinement is important to obtain reliable results. A mesh refinement study has been performed in the Abaqus reference model, illustrated in Figure 3.5a. For comparison the solution obtained using C3D8I elements and 40 elements in the longitudinal direction has been taken as reference, as it proved to give the closest results to the analytical solution [58]. On the other hand, it can be observed how the geometric stiffening effect becomes clearly visible in C3D8 elements using at least 40 elements along the length of the beam. The same behavior has been observed also in the full integration CR user element, where the deformed shapes obtained with different mesh seeds are reported in Figure 3.5b.



(a) Mesh refinement study

(b) Geometric non-linear cantilever beam deformed shape

Figure 3.5: Cantilever beam model: influence of mesh seeds in the longitudinal direction

Using only one element through the thickness, a stiff element behavior is observed due to the shear locking phenomenon. In order to alleviate this effect, a selective integration scheme has been implemented. According to this scheme, a full integration is used for the stress calculation of all terms except the transverse shear terms, for which a reduced integration scheme is used. The results are shown in Figure 3.6. From the comparison it can be observed that the full integration CR element has a slightly stiffer response when compared to the Abaqus C3D8 element. However, it is worth reminding that also the linear UEL has been found to be slightly stiffer compared to the C3D8 standard element, as reported in Section 2.3. On the other hand, using the selective integration scheme improves the results, that are in close agreement with the C3D8I elements taken as reference.

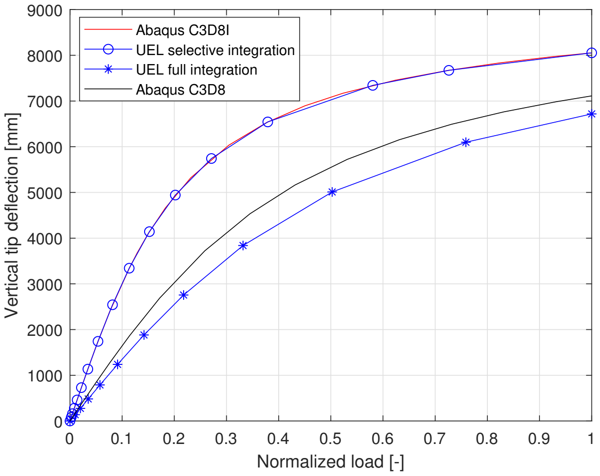


Figure 3.6: Output results for the beam subjected to end shear load

Regarding the element performance, it has been observed that the CR full integration element takes 55 iterations, exactly as the Abaqus C3D8 standard element, but requires less increments to complete the analysis (15 compared to 21). At the same time, the CR element proved to be more stable, as there were less warning messages during the analysis. Using selective integration, the CR element requires 16 increments with 64 iterations, compared to the 21 increments with 59 iterations for the C3D8R standard element. On the other hand, the computational time required to complete the analysis has been found to be much higher for the user element compared to Abaqus standard elements.

3.3.2. Beam subjected to end moment load

In this model, which is reported by Sze et al. [56], a cantilever beam is subjected to an end moment load, until it is bent into a circle. This one is illustrated in Figure 3.7. The beam has fixed boundary conditions at the right end, while a 2π rotation is imposed on a reference node coupled to the left end. An isotropic material with elastic modulus $\mathrm{E} = 1.2\mathrm{MPa}$ and Poisson ratio $\nu = 0$ is used for this model. The beam has length $\ell = 12\mathrm{m}$, width $\mathrm{b} = 1\mathrm{m}$ and thickness $\mathrm{h} = 0.1\mathrm{m}$.

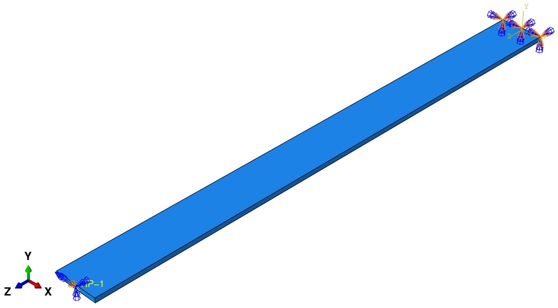
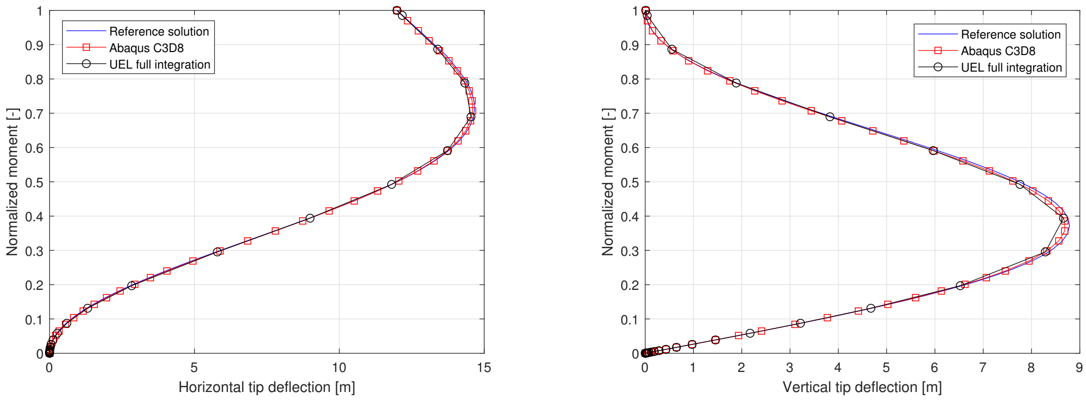


Figure 3.7: Reference model for the cantilever beam subjected to an end moment

In order to draw a comparison between the Abaqus C3D8 element and the CR UEL, the end moment is plotted against the vertical and horizontal tip deflections in Figure 3.8. As it can be observed, there is very close agreement between the CR element and the reference solution. In this benchmark, the selective integration scheme has not been used, as the solution obtained with the full integration scheme was found to be already in good agreement with the reference. Concerning the element performance, a considerable computational saving has been observed in this model. Using the CR element, the analysis completes in 27 increments with 126 iterations. On the other hand, by using the standard C3D8 element, the analysis requires 52 increments and 221 iterations. The deformed shape of the beam throughout the analysis is illustrated in Figure 3.9.



(a) Horizontal tip displacement

(b) Vertical tip displacement

Figure 3.8: Output results for the beam subjected to end moment load

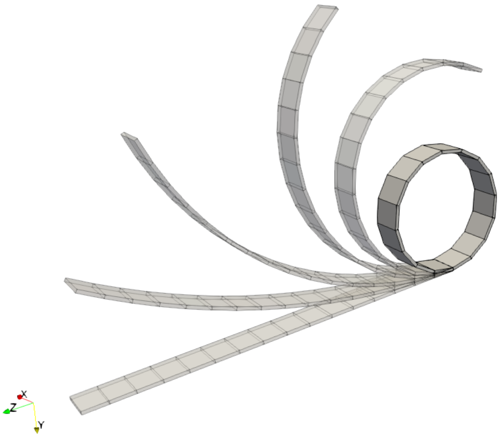


Figure 3.9: Deformed shape of the cantilever beam throughout the analysis

3.3.3. Slit ring subjected to a line force

This model has been extensively used as benchmark for geometric non-linear codes and has also been reported by Sze et al. [56]. The model is illustrated in Figure 3.10.

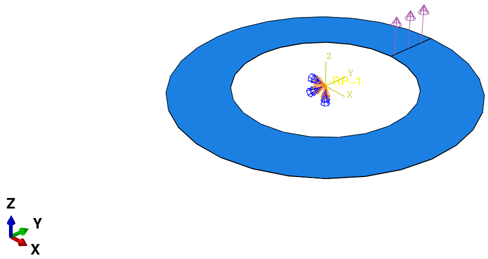
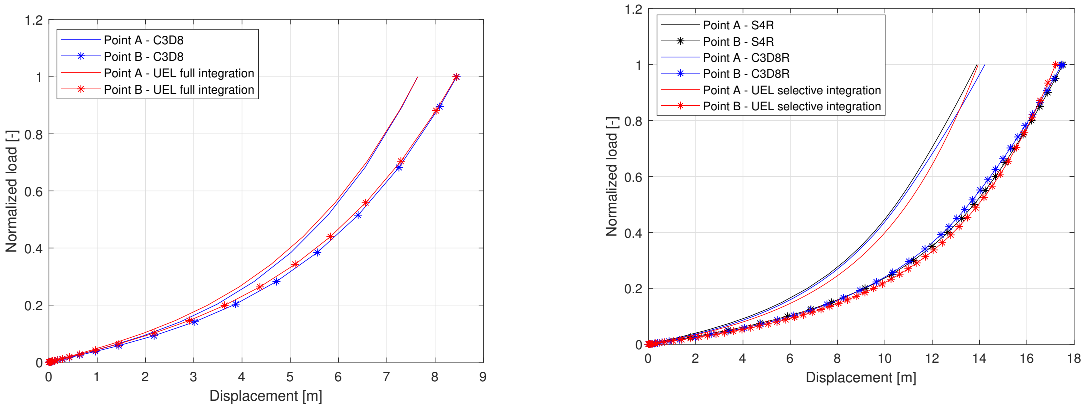


Figure 3.10: Reference model for the slit ring subjected to a line force

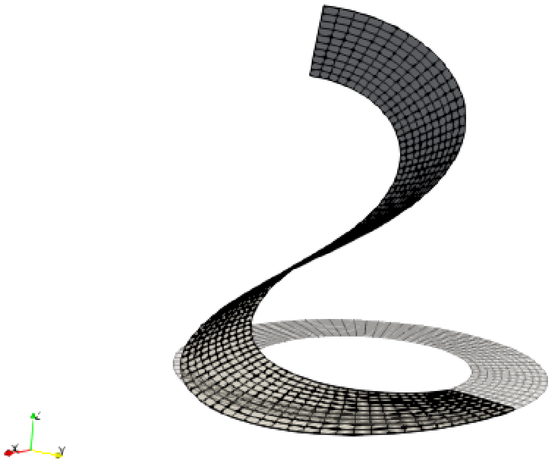
The ring has an internal radius $\mathrm{R_i} = 6\mathrm{m}$, an external radius $\mathrm{R_e} = 10\mathrm{m}$ and height $\mathrm{h} = 0.03\mathrm{m}$. An isotropic material with elastic modulus $\mathrm{E} = 21\mathrm{MPa}$ and Poisson ratio $\nu = 0$ has been assigned to the part. The ring is characterized by a small cut, so that two ends are present. One end is clamped and it is kinematically coupled to a reference node located at the center of the ring, while a line load $\mathrm{P} = 0.8\mathrm{N/m}$ is applied to the other end. Denoting with A and B the points at the innermost and outermost locations of the loaded end, respectively, the interest is in determining the large rotation response of the ring by tracking the vertical displacement of these points.

The model was meshed with 10 elements in the radial direction and 80 in the circumferential direction. The comparison with Abaqus standard elements is reported in Figure 3.11. As it can be observed in Figure 3.11a, there is good agreement between the CR full integration element and the C3D8 Abaqus element. However, both of them are not close to the reference solution, which is provided by S4R elements [56], due to shear locking . As a result, the selective integration scheme has been used. The results are compared in Figure 3.11b, where it can be observed that there is close agreement with the reference. Considering the element performance in this model, the CR full integration element completed the analysis in 22 increments, with 184 iterations, compared to 21 increments and 55 iterations for the C3D8 element. The deformed shape for $\mathrm{P} = \mathrm{P_{max}}$ can be visualized in Figure 3.11c. Using the selective integration scheme, the analysis completed in 55 increments with 4093 iterations, while it required 43 increments and 260 iterations for the Abaqus C3D8R element.



(a) Comparison between C3D8 and UEL full integration

(b) Comparison between C3D8R and UEL selective integration



(c) Deformed shape of the slit ring subjected to a line force

Figure 3.11: Output results for the slit ring subjected to a line force

4

The Floating Node Method

In this chapter, a comprehensive overview on the Floating Node Method (FNM) and its combination with the CR formulation is presented. Section 4.1 introduces the reader to the concept of *floating node*, *enriched solid ply element* and *enriched cohesive interface element*. These elements are subsequently used to construct an *enriched laminate element*. Next, the section gives a detailed explanation of the FNM partitioning algorithm, through which it is possible to explicitly represent multiple matrix cracks and widespread delamination in a composite laminate. Damage initiation/propagation criteria used in this method are described in Section 4.2. Finally, the integration of the FN element in the co-rotational approach is presented in Section 4.3, where some test models are reported to show the element's potential in addressing geometric non-linear fracture problems in composites.

4.1. Overview of the approach

In Section 2.3, the implementation of a standard three-dimensional finite element has been discussed. Even though we can identify several geometric features, it might have been noticed that, in a FEM framework, the element is merely defined through its physical nodes, its nodal connectivity and associated shape functions. The reason behind is that the domain is assumed to be continuous throughout the analysis, so that only the solution values at the physical nodes are needed. However, in order to have an accurate representation of the crack geometry, it is crucial to not impose any a priori restriction on the crack initiation location and crack propagation direction, as this is known to follow the most energetically favourable path. This is the key motivation that lead to the formulation of the so called *enriched methods*, which allow discontinuities to be modelled inside the element domain. When a crack propagates inside an element, new DOFs and shape functions are required to represent its boundaries and interpolate the solution over its physical nodes to track its motion.

In order to allow crack propagation inside the element domain, the FNM uses an *enriched* element definition. While maintaining the same nodal connectivity, the single FN element is able to split itself into two or more *sub-elements* (SE) using information about its edge connectivity and by introduction of an appropriate number of *floating nodes*. The floating nodes are so called because they are not assigned any position vector at the beginning of the analysis. However, when a fracture criterion is met, these will assume a specific location along the element edges cut by the crack to form the crack boundaries to represent the displacement solution at the crack nodes.

Considering the complexity of fracture modelling in composites, the enriched element formulation is applied to obtain *ply* and *cohesive interface elements*, purposely designed to model matrix cracks, delamination and the interaction between the two. Furthermore, these ones are embedded in a single *FN laminate element*, which allows different ply and interface SEs within such element to share information with each other. The implementation procedure outlined in the following sections is based on the work of Chen et al. [28, 34, 50, 51, 66, 67].

4.1.1. FN ply element definition

The FN ply element is an 8-node hexahedral element with enriched topological definition. An FN ply element is purposely designed to represent a matrix crack within its initially intact domain. The element definition is illustrated in Figure 4.1.

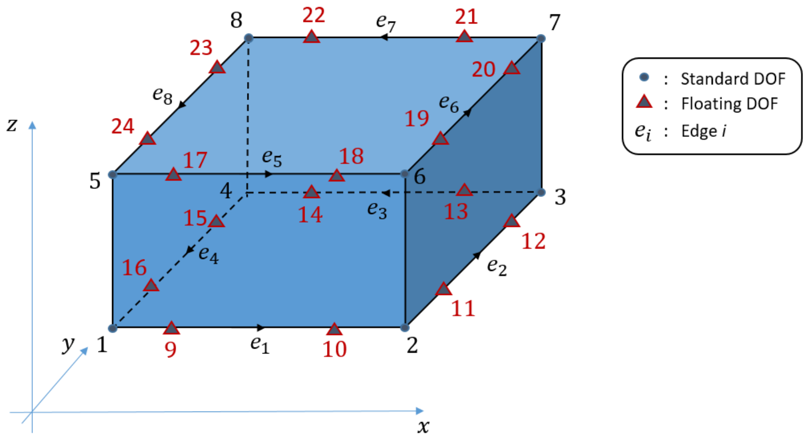


Figure 4.1: Topological definition of the FN ply element

As it can be observed, the element is characterized by 24 nodes, 8 of which are the standard nodes, while the other 16 are floating nodes. Furthermore, an edge list is created for the element, where the bottom and top surface edges are numbered according to Figure 4.1. These ones are the only relevant edges, as it is assumed that the matrix crack is perpendicular to the top and bottom surface of the element. The floating DOFs are initially associated to such edges.

If the FN element is intact, this one will behave as a standard hexahedral element. Denoting with Ω the physical domain and with Γ_t the boundary of such domain, the discretized system of equilibrium equations for the element, derived in Section 2.2.1 but reported here for convenience, is given by:

$$\int_\Omega \boldsymbol{\epsilon}^{\mathrm{T}}(\delta\mathbf{v})\boldsymbol{\sigma}\,\mathrm{d}\Omega = \int_\Omega \delta\mathbf{v}^{\mathrm{T}}\rho\mathbf{b}\,\mathrm{d}\Omega + \int_{\Gamma_t}\delta\mathbf{v}^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma \tag{4.1.1.1}$$

As a result, the same element calculations described in Section 2.3 are performed, where the only difference is in the number of DOFs, as the element has now 16 additional floating nodes. However, in such a case, the floating DOFs are not yet active and therefore useless. Since it is important to avoid singularities in the stiffness matrix to help convergence, the diagonal entries corresponding to floating DOFs can be assigned a dummy stiffness value, or it can be decided to remove the rows and columns corresponding to these DOFs. Hence, by defining the element position vector $\mathbf{x}_\Omega$ and the displacement solution $\mathbf{u}$ as:

$$\mathbf{x}_\Omega = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_8]^{\mathrm{T}}; \qquad \mathbf{u}(\mathbf{x}) = \mathbf{N}(\mathbf{x})\mathbf{u}_\Omega \tag{4.1.1.2}$$

It is possible to write Equation 4.1.1.1 as:

$$\mathbf{K}_\Omega\mathbf{u}_\Omega = \mathbf{f}_\Omega^{\mathrm{ext}} \tag{4.1.1.3}$$

Where $\mathbf{K}_\Omega$ and $\mathbf{f}_\Omega^{\mathrm{ext}}$ are defined by:

$$\mathbf{K}_\Omega = \int_\Omega \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}\,\mathrm{d}\Omega; \qquad \mathbf{f}_\Omega^{\mathrm{ext}} = \int_\Omega \mathbf{N}^{\mathrm{T}}\rho\mathbf{b}\,\mathrm{d}\Omega + \int_{\Gamma_t}\mathbf{N}^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma \tag{4.1.1.4}$$

On the other hand, if a certain fracture criterion is met, the element domain is partitioned. Let us assume $\mathrm{e}_1, \mathrm{e}_3, \mathrm{e}_5, \mathrm{e}_7$ being the four broken edges cut by the matrix crack, as illustrated in Figure 4.2.

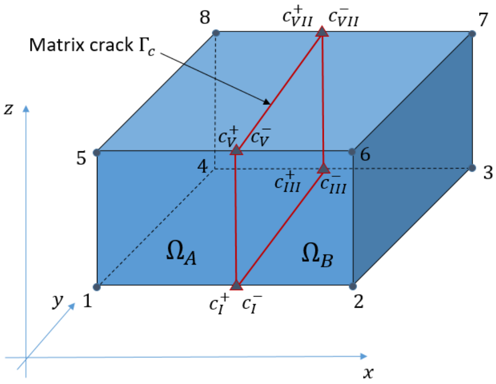


Figure 4.2: Partitioning of the FN ply element due to matrix crack

In this case, the matrix crack creates four pairs of initially coincident nodes, $\mathrm{c}_{\mathrm{I}}^{\pm}, \mathrm{c}_{\mathrm{III}}^{\pm}, \mathrm{c}_{\mathrm{V}}^{\pm}, \mathrm{c}_{\mathrm{VII}}^{\pm}$, on the four edges respectively. The coordinates of these nodes are obtained by the intersection of the matrix crack with the four edges, as it is known that the matrix crack propagates along the fiber direction. The floating DOFs allocated to the edges at the beginning of the analysis are in such case used to represent the displacements at the crack nodes:

$$\begin{aligned}
&\mathbf{u}_{\mathrm{c}_{\mathrm{I}}^{+}} = \mathbf{u}_9; && \mathbf{u}_{\mathrm{c}_{\mathrm{III}}^{+}} = \mathbf{u}_{14}; && \mathbf{u}_{\mathrm{c}_{\mathrm{V}}^{+}} = \mathbf{u}_{17}; && \mathbf{u}_{\mathrm{c}_{\mathrm{VII}}^{+}} = \mathbf{u}_{22};\\
&\mathbf{u}_{\mathrm{c}_{\mathrm{I}}^{-}} = \mathbf{u}_{10}; && \mathbf{u}_{\mathrm{c}_{\mathrm{III}}^{-}} = \mathbf{u}_{13}; && \mathbf{u}_{\mathrm{c}_{\mathrm{V}}^{-}} = \mathbf{u}_{18} && \mathbf{u}_{\mathrm{c}_{\mathrm{VII}}^{-}} = \mathbf{u}_{21}
\end{aligned} \tag{4.1.1.5}$$

Where $\mathbf{u}_{\mathrm{c}_{\mathrm{i}}^{\pm}}$ denotes the displacements at the crack node i. With the presence of the matrix crack, the original element domain Ω is divided into two sub-domains Ω_{A} and Ω_{B}. Furthermore, denoting with Γ_{c} the crack boundaries, assuming that the crack is a cohesive crack, the discretized system of equilibrium equations for the element becomes:

$$\int_{\Omega_{\mathrm{A}}\cup\Omega_{\mathrm{B}}} \boldsymbol{\epsilon}^{\mathrm{T}}(\delta\mathbf{v})\boldsymbol{\sigma}(\mathbf{u})\,\mathrm{d}\Omega + \int_{\Gamma_{\mathrm{c}}} [\![\delta\mathbf{v}]\!]^{\mathrm{T}}\boldsymbol{\tau}_{\mathrm{c}}([\![\mathbf{u}]\!])\mathrm{d}\Gamma = \int_{\Omega_{\mathrm{A}}\cup\Omega_{\mathrm{B}}} \delta\mathbf{v}^{\mathrm{T}}\rho\mathbf{b}\,\mathrm{d}\Omega + \int_{\Gamma_{\mathrm{t}}} \delta\mathbf{v}^{\mathrm{T}}\bar{\mathbf{t}}\,\mathrm{d}\Gamma \tag{4.1.1.6}$$

Where $[\![\bullet]\!]$ represents the discontinuity jump of a function across the crack surfaces and $\boldsymbol{\tau}_{\mathrm{c}}$ is the traction resisting the separation $[\![\mathbf{u}]\!]$ between the cohesive surfaces, related to the latter through the following:

$$\boldsymbol{\tau}_{\mathrm{c}} = \mathbf{D}_{\mathrm{CE}}[\![\mathbf{u}]\!] \tag{4.1.1.7}$$

Where $\mathbf{D}_{\mathrm{CE}}$ incorporates the cohesive constitutive law, given in Section A.2. Therefore, the undamaged element is partitioned into three SEs, two hexahedral elements and a cohesive element representing the matrix crack, utterly defined by floating nodes. The nodal position vectors of the three elements are:

$$\begin{aligned}
\mathbf{x}_{\Omega_{\mathrm{A}}} &= [\mathbf{x}_1, \mathbf{x}_{\mathrm{c}_{\mathrm{I}}^{+}}, \mathbf{x}_{\mathrm{c}_{\mathrm{III}}^{+}}, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_{\mathrm{c}_{\mathrm{V}}^{+}}, \mathbf{x}_{\mathrm{c}_{\mathrm{VII}}^{+}}, \mathbf{x}_8]^{\mathrm{T}}\\
\mathbf{x}_{\Omega_{\mathrm{B}}} &= [\mathbf{x}_{\mathrm{c}_{\mathrm{I}}^{-}}, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_{\mathrm{c}_{\mathrm{III}}^{-}}, \mathbf{x}_{\mathrm{c}_{\mathrm{V}}^{-}}, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_{\mathrm{c}_{\mathrm{VII}}^{-}}]^{\mathrm{T}}\\
\mathbf{x}_{\Gamma_{\mathrm{c}}} &= [\mathbf{x}_{\mathrm{c}_{\mathrm{I}}^{-}}, \mathbf{x}_{\mathrm{c}_{\mathrm{V}}^{-}}, \mathbf{x}_{\mathrm{c}_{\mathrm{VII}}^{-}}, \mathbf{x}_{\mathrm{c}_{\mathrm{III}}^{-}}, \mathbf{x}_{\mathrm{c}_{\mathrm{I}}^{+}}, \mathbf{x}_{\mathrm{c}_{\mathrm{V}}^{+}}, \mathbf{x}_{\mathrm{c}_{\mathrm{VII}}^{+}}, \mathbf{x}_{\mathrm{c}_{\mathrm{III}}^{+}}]^{\mathrm{T}}
\end{aligned} \tag{4.1.1.8}$$

On the other hand, the nodal displacements associated with the elements are:

$$\begin{aligned}
\mathbf{u}_{\Omega_{\mathrm{A}}} &= [\mathbf{u}_1, \mathbf{u}_9, \mathbf{u}_{14}, \mathbf{u}_4, \mathbf{u}_5, \mathbf{u}_{17}, \mathbf{u}_{22}, \mathbf{u}_8]^{\mathrm{T}}\\
\mathbf{u}_{\Omega_{\mathrm{B}}} &= [\mathbf{u}_{10}, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_{13}, \mathbf{u}_{18}, \mathbf{u}_6, \mathbf{u}_7, \mathbf{u}_{21}]^{\mathrm{T}}\\
\mathbf{u}_{\Gamma_{\mathrm{c}}} &= [\mathbf{u}_{10}, \mathbf{u}_{18}, \mathbf{u}_{21}, \mathbf{u}_{13}, \mathbf{u}_9, \mathbf{u}_{17}, \mathbf{u}_{22}, \mathbf{u}_{14}]^{\mathrm{T}}
\end{aligned} \tag{4.1.1.9}$$

Hence, the displacement solution can be interpolated over the bulk SEs nodes using the standard shape functions for hexahedral elements:

$$\begin{aligned} \mathbf{u}(\mathbf{x}) &= \mathbf{N}_{\Omega_\mathrm{A}}(\mathbf{x})\mathbf{u}_{\Omega_\mathrm{A}} && \text{for } \mathbf{x} \in \Omega_\mathrm{A} \\ &= \mathbf{N}_{\Omega_\mathrm{B}}(\mathbf{x})\mathbf{u}_{\Omega_\mathrm{B}} && \text{for } \mathbf{x} \in \Omega_\mathrm{B} \end{aligned} \qquad (4.1.1.10)$$

Where $\mathbf{N}_{\Omega_\mathrm{A}}$ and $\mathbf{N}_{\Omega_\mathrm{B}}$ are the shape functions matrices for the two SEs, expressed in physical coordinates. On the other hand, the displacement jump between the cohesive surfaces is related to the nodal displacements $\mathbf{u}_{\Gamma_\mathrm{c}}$ through the standard shape functions for cohesive elements $\mathbf{N}_{\Gamma_\mathrm{c}}$, as reported in Section A.2:

$$[\![\mathbf{u}]\!](\mathbf{x}) = \mathbf{N}_{\Gamma_\mathrm{c}}(\mathbf{x})\mathbf{u}_{\Gamma_\mathrm{c}} \qquad \text{for } \mathbf{x} \in \Gamma_\mathrm{c} \qquad (4.1.1.11)$$

As a result, the terms on the LHS of Equation 4.1.1.6 lead to the definition of the SEs stiffness matrices:

$$\begin{aligned} \mathbf{K}_{\Omega_\mathrm{A}} &= \int_{\Omega_\mathrm{A}} \mathbf{B}_\mathrm{A}^\mathrm{T}\mathbf{D}\mathbf{B}_\mathrm{A}\,\mathrm{d}\Omega \\ \mathbf{K}_{\Omega_\mathrm{B}} &= \int_{\Omega_\mathrm{B}} \mathbf{B}_\mathrm{B}^\mathrm{T}\mathbf{D}\mathbf{B}_\mathrm{B}\,\mathrm{d}\Omega \\ \mathbf{K}_{\Gamma_\mathrm{c}} &= \int_{\Gamma_\mathrm{c}} \mathbf{N}_{\Gamma_\mathrm{c}}^\mathrm{T}\mathbf{D}_\mathrm{CE}\mathbf{N}_{\Gamma_\mathrm{c}}\,\mathrm{d}\Gamma \end{aligned} \qquad (4.1.1.12)$$

Where $\mathbf{B}_\mathrm{A}$ and $\mathbf{B}_\mathrm{B}$ are the strain-displacement matrices for the bulk SEs. It is worth noting that each bulk SE has its own Jacobian matrix as opposed to the PNM. Assuming no external loading on the crack surfaces ($\mathbf{f}_{\Gamma_\mathrm{c}}^\mathrm{ext} = \mathbf{0}$), the external force vectors for the bulk SEs are given by:

$$\begin{aligned} \mathbf{f}_{\Omega_\mathrm{A}}^\mathrm{ext} &= \int_{\Omega_\mathrm{A}} \mathbf{N}_{\Omega_\mathrm{A}}^\mathrm{T}\rho\mathbf{b}\,\mathrm{d}\Omega + \int_{\Gamma_\mathrm{t}\cap\Gamma_{\Omega_\mathrm{A}}} \mathbf{N}_{\Omega_\mathrm{A}}^\mathrm{T}\bar{\mathbf{t}}\,\mathrm{d}\Gamma \\ \mathbf{f}_{\Omega_\mathrm{B}}^\mathrm{ext} &= \int_{\Omega_\mathrm{B}} \mathbf{N}_{\Omega_\mathrm{B}}^\mathrm{T}\rho\mathbf{b}\,\mathrm{d}\Omega + \int_{\Gamma_\mathrm{t}\cap\Gamma_{\Omega_\mathrm{B}}} \mathbf{N}_{\Omega_\mathrm{B}}^\mathrm{T}\bar{\mathbf{t}}\,\mathrm{d}\Gamma \end{aligned} \qquad (4.1.1.13)$$

However, an external load may be introduced on the crack surface by simply apply the desired loading directly on the floating nodes at the beginning of the analysis. Finally, the system of discretized equilibrium equations for the initial element is obtained through assembly of the SEs contributions to the system of equations. Again, in order to avoid singularities, upon assembly it can be decided to either remove the contribution of non active floating DOFs or to assign dummy stiffness values to their corresponding diagonal entries.

In the above illustrated cracking scenario, it has been assumed that the element is partitioned into two hexahedral SEs. However, depending on the fibers orientation in the element, the matrix crack can be such that a triangular prism and a pentagonal prism are created, as illustrated in Figure 4.3. The pentagonal prism can in turn be considered to be composed of three triangular prisms. As wedge elements are supported as potential SE in the FNM, this and many other cracking scenarios can be captured. It is worth noting that, after the original element is partitioned, the system of discretized equilibrium equations to be solved is equivalent to a standard FEM mesh representing explicitly the partition.

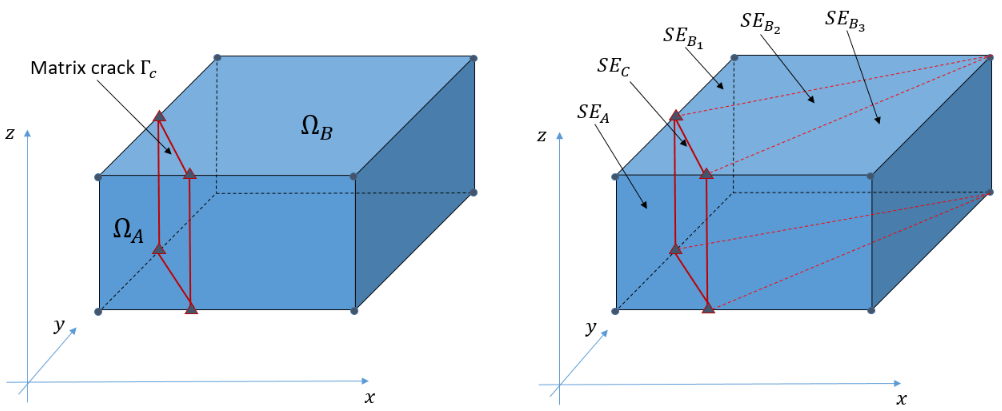


Figure 4.3: Partitioning of a pentagonal prism into three wedge elements to perform numerical integration

4.1.2. FN cohesive element definition

In this section, the formulation of the FN cohesive element with enriched topological definition is presented. The FN cohesive element is characterized by an evolving topology as the FN ply element, such that it is able to represent the boundaries of a crack explicitly, capturing the stress concentration at the crack tip. The enriched cohesive element topology is given in Figure 4.4.

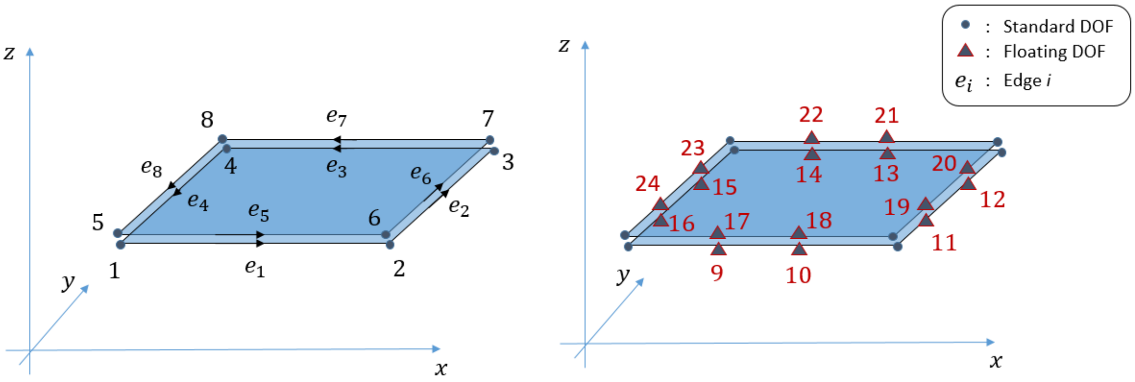


Figure 4.4: Topological definition of the FN cohesive element

Similarly to the FN ply element, the FN cohesive element is defined by 8 physical nodes and 16 floating nodes, together with a list of all its edges. In addition to these, other 8 *internal nodes* are introduced in the element definition, the purpose of which will be explained later. If the FN cohesive element is intact, this one will behave as a standard 8-node cohesive element, described in Section A.2. Similarly to the FN ply element, since the floating nodes are not active, the contribution of the floating DOFs is removed from the system of equations. Therefore, the nodal position vector of the intact FN element and the nodal displacement vector are given by:

$$\begin{aligned}\mathbf{x}_{\mathrm{CE}} &= [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_8]^{\mathrm{T}} \\ \mathbf{u}_{\mathrm{CE}} &= [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_8]^{\mathrm{T}}\end{aligned} \tag{4.1.2.1}$$

The displacement jump between the top and bottom surface of the cohesive element in global coordinates is related to the nodal displacements through the following:

$$[\![\mathbf{u}]\!](\mathbf{x}) = \mathbf{N}_{\mathrm{CE}}(\mathbf{x})\mathbf{u}_{\mathrm{CE}} \tag{4.1.2.2}$$

Denoting with Γ_{CE} the domain of the cohesive element, the stiffness matrix of the cohesive element is given by:

$$\mathbf{K}_{\mathrm{CE}} = \int_{\Gamma_{\mathrm{CE}}} \mathbf{N}_{\mathrm{CE}}^{\mathrm{T}} \mathbf{D}_{\mathrm{CE}} \mathbf{N}_{\mathrm{CE}} \, \mathrm{d}\Gamma \tag{4.1.2.3}$$

The element calculations are in this case performed in a standard way, as outlined in Section A.2.

On the other hand, if a certain fracture criterion is met, the cohesive element domain undergoes partitioning. Let us assume the top surface of the cohesive element to be cut by a matrix crack, with the crack nodes located on the edges e_5 and e_7 respectively, as illustrated in Figure 4.5.

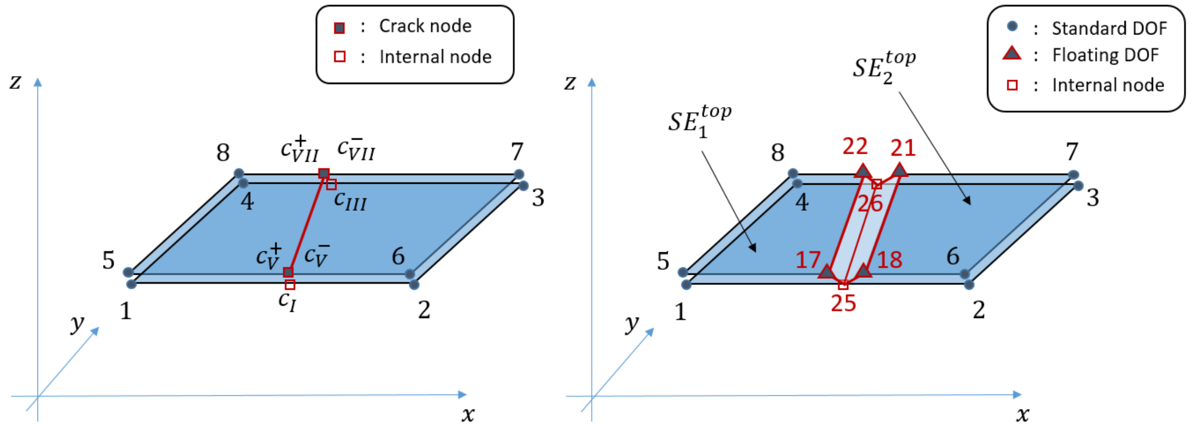


Figure 4.5: Partitioning of the FN cohesive element due to a matrix crack on its top surface

In this case, two pairs of initially coincident crack nodes are created, denoted with $\mathrm{c}_{\mathrm{V}}^{\pm}$ and $\mathrm{c}_{\mathrm{VII}}^{\pm}$. At the same time, two extra internal nodes, initially coincident with the previously defined crack nodes, are located at the bottom surface of the cohesive element, c_{I} and $\mathrm{c}_{\mathrm{III}}$ respectively. With these nodes defined, the initial element is partitioned into two SEs, $\mathrm{SE}_1^{\mathrm{top}}$ and $\mathrm{SE}_2^{\mathrm{top}}$, with the following nodal position vectors:

$$\mathbf{x}_{\mathrm{SE}_1^{\mathrm{top}}} = [\mathbf{x}_1, \mathbf{x}_{\mathrm{c}_{\mathrm{I}}}, \mathbf{x}_{\mathrm{c}_{\mathrm{III}}}, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_{\mathrm{c}_{\mathrm{V}}^{+}}, \mathbf{x}_{\mathrm{c}_{\mathrm{VII}}^{+}}, \mathbf{x}_8]^{\mathrm{T}} \tag{4.1.2.4}$$

$$\mathbf{x}_{\mathrm{SE}_2^{\mathrm{top}}} = [\mathbf{x}_{\mathrm{c}_{\mathrm{I}}}, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_{\mathrm{c}_{\mathrm{III}}}, \mathbf{x}_{\mathrm{c}_{\mathrm{V}}^{-}}, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_{\mathrm{c}_{\mathrm{VII}}^{-}}]^{\mathrm{T}} \tag{4.1.2.5}$$

Similarly to the FN ply element, the floating DOFs allocated to the edges at the beginning of the analysis are used to represent the displacements at the crack nodes, while the internal auxiliary nodes are used to represent the crack tip:

$$\begin{aligned} \mathbf{u}_{\mathrm{c}_{\mathrm{V}}^{+}} &= \mathbf{u}_{17}; & \mathbf{u}_{\mathrm{c}_{\mathrm{V}}^{-}} &= \mathbf{u}_{18}; & \mathbf{u}_{\mathrm{c}_{\mathrm{VII}}^{+}} &= \mathbf{u}_{22} \\ \mathbf{u}_{\mathrm{c}_{\mathrm{VII}}^{-}} &= \mathbf{u}_{21}; & \mathbf{u}_{\mathrm{c}_{\mathrm{I}}} &= \mathbf{u}_{25}; & \mathbf{u}_{\mathrm{c}_{\mathrm{III}}} &= \mathbf{u}_{26} \end{aligned} \tag{4.1.2.6}$$

However, since the bottom edges, where the nodes representing the crack tip are located, are still intact, an interpolation of the displacement solution is needed to obtain the nodal displacements of these nodes:

$$\begin{aligned}\mathbf{u}_{25} &= \mathrm{N}_1(\mathbf{x}_{\mathrm{cI}})\mathbf{u}_1 + \mathrm{N}_2(\mathbf{x}_{\mathrm{cI}})\mathbf{u}_2 \\ \mathbf{u}_{26} &= \mathrm{N}_3(\mathbf{x}_{\mathrm{cIII}})\mathbf{u}_3 + \mathrm{N}_4(\mathbf{x}_{\mathrm{cIII}})\mathbf{u}_4\end{aligned} \tag{4.1.2.7}$$

Hence, the nodal displacement vectors for the SEs are defined as:

$$\begin{aligned}\mathbf{u}_{\mathrm{SE}_1^{\mathrm{top}}} &= [\mathbf{u}_1, \mathbf{u}_{25}, \mathbf{u}_{26}, \mathbf{u}_4, \mathbf{u}_5, \mathbf{u}_{17}, \mathbf{u}_{22}, \mathbf{u}_8]^{\mathrm{T}} \\ \mathbf{u}_{\mathrm{SE}_2^{\mathrm{top}}} &= [\mathbf{u}_{25}, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_{26}, \mathbf{u}_{18}, \mathbf{u}_6, \mathbf{u}_7, \mathbf{u}_{21}]^{\mathrm{T}}\end{aligned}$$

The displacement jump across the surfaces of the two SEs is given by:

$$\begin{aligned}[\![\mathbf{u}]\!](\mathbf{x}) &= \mathbf{N}_{\mathrm{SE}_1^{\mathrm{top}}}(\mathbf{x})\mathbf{u}_{\mathrm{SE}_1^{\mathrm{top}}} && \text{for } \mathbf{x} \in \mathrm{SE}_1^{\mathrm{top}} \\ &= \mathbf{N}_{\mathrm{SE}_2^{\mathrm{top}}}(\mathbf{x})\mathbf{u}_{\mathrm{SE}_2^{\mathrm{top}}} && \text{for } \mathbf{x} \in \mathrm{SE}_2^{\mathrm{top}}\end{aligned} \tag{4.1.2.8}$$

Where $\mathbf{N}_{\mathrm{SE}_1^{\mathrm{top}}}$ and $\mathbf{N}_{\mathrm{SE}_2^{\mathrm{top}}}$ are the shape function matrices for the two SEs expressed in physical coordinates. The stiffness matrices of the two SEs are given by:

$$\begin{aligned}\mathbf{K}_{\mathrm{SE}_1^{\mathrm{top}}} &= \int_{\Gamma_{\mathrm{SE}_1^{\mathrm{top}}}} \mathbf{N}^{\mathrm{T}}_{\mathrm{SE}_1^{\mathrm{top}}}\mathbf{D}_{\mathrm{CE}}\mathbf{N}_{\mathrm{SE}_1^{\mathrm{top}}}\,\mathrm{d}\Gamma \\ \mathbf{K}_{\mathrm{SE}_1^{\mathrm{top}}} &= \int_{\Gamma_{\mathrm{SE}_2^{\mathrm{top}}}} \mathbf{N}^{\mathrm{T}}_{\mathrm{SE}_2^{\mathrm{top}}}\mathbf{D}_{\mathrm{CE}}\mathbf{N}_{\mathrm{SE}_2^{\mathrm{top}}}\,\mathrm{d}\Gamma\end{aligned} \tag{4.1.2.9}$$

Since the internal auxiliary nodes are located on intact edges, it is necessary to remove their DOFs from the system of equations. This is achieved by using the relations given in Equation 4.1.2.7, so that their contribution is condensed onto the DOFs of the end nodes of the intact edges. If damage is already present in the element before the partition, the damage variables are passed to the SEs and are used to degrade the cohesive constitutive law.

In the above considered cracking scenario, the matrix crack was assumed to be located at the top surface of the cohesive element. However, if it is located at the bottom surface, the same procedure is followed so that two SEs, $\mathrm{SE}_1^{\mathrm{bot}}$ and $\mathrm{SE}_2^{\mathrm{bot}}$, are created. Another possible case is the one in which both the top and bottom surface of the cohesive element contain a matrix crack. In such circumstances, both the top SE and the bottom SE are integrated separately. However, following this procedure will result in the element being integrated twice over its original domain. As a result, the final solution is assumed to be a weighted average of the two SEs:

$$\begin{aligned}\mathbf{K}_{\mathrm{CE}} &= \frac{1}{2}\Big(\mathbf{K}_{\mathrm{SE}^{\mathrm{top}}} + \mathbf{K}_{\mathrm{SE}^{\mathrm{bot}}}\Big) \\ \mathbf{f}^{\mathrm{int}}_{\mathrm{CE}} &= \frac{1}{2}\Big(\mathbf{f}^{\mathrm{int}}_{\mathrm{SE}^{\mathrm{top}}} + \mathbf{f}^{\mathrm{int}}_{\mathrm{SE}^{\mathrm{bot}}}\Big)\end{aligned}$$

4.1.3. FN laminate element definition

With the FN ply and FN cohesive elements defined, these are used as SEs to give birth to a FN laminate element. Let us consider a two-ply laminate element as the one illustrated in Figure 4.6. This laminate element will contain two enriched ply SEs and one enriched cohesive SE representing the interface between the plies.

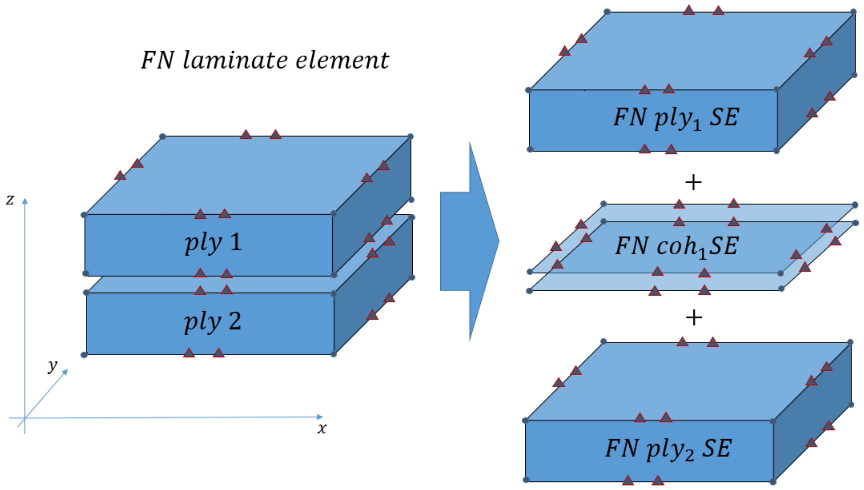


Figure 4.6: Construction of the FN laminate element

In order to define the FN laminate element connectivity, a Python script is used for the pre-processing of the Abaqus input file. The Python script takes as input an Abaqus input file, where the part to be meshed with FN laminate elements is meshed with standard C3D8 elements, oriented such that all the elements are laying on the xy plane and having one element only in the thickness direction, i.e. the z direction. The part is assigned a specific keyword in order to be recognized by the Python script. The script prepares a new Abaqus input file where the part is meshed with FN laminate elements and generates relevant data-lists to be read by the UEL through the UEXTERNALDB subroutine. By specification of the stacking sequence for the laminate and the single ply thickness, the new nodal coordinates for the ply and interface SEs are calculated.

At the same time, the DOFs for the laminate element are allocated as follows. Let us denote with $\mathrm{N_{DOF,ply}}$ the DOFs of a single FN ply element and with $\mathrm{N_{DOF,coh}}$ the DOFs of a single FN cohesive element. If the FN laminate element contains n plies, its total number of DOFs is given by:

$$\mathrm{N_{DOF,lam}} = \mathrm{n} \cdot \mathrm{N_{DOF,ply}} \tag{4.1.3.1}$$

The $\mathrm{i-th}$ ply SE has its j DOF allocated in its $\mathrm{array_{DOF,ply_i}}$ DOF array according to:

$$\mathrm{array_{DOF,ply_i}(j)} = \mathrm{(i-1)} \cdot \mathrm{N_{DOF,ply}} + \mathrm{j} \qquad \text{with } \mathrm{j} = 1, 2, \ldots, \mathrm{N_{DOF,ply}} \tag{4.1.3.2}$$

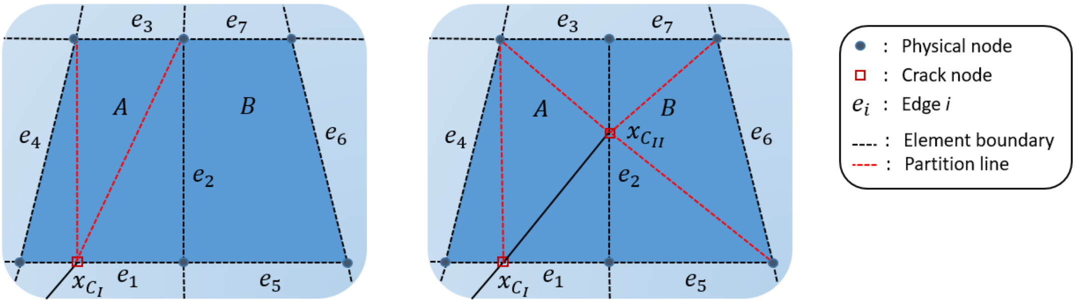
Considering the $\mathrm{i-th}$ cohesive SE, let us denote with $\mathrm{N_{DOF,coh}}$ its number of DOFs. As the FN laminate element is assembled from top to bottom, the $\mathrm{i-th}$ cohesive SE has its top surface DOFs coincident with the bottom surface of the $\mathrm{i-th}$ ply, and its bottom surface DOFs coincident with the top surface of the $\mathrm{(i+1)-th}$ ply. Hence its DOF array $\mathrm{array_{DOF,coh_i}}$ is given by:

$$\begin{aligned} \mathrm{array_{DOF,coh_i}(k)} &= \mathrm{array_{DOF,ply_i}(k + N_{DOF,ply})} \\ \mathrm{array_{DOF,coh_i}(k + N_{DOF,coh}/2)} &= \mathrm{array_{DOF,ply_{i+1}}(k)} \end{aligned} \tag{4.1.3.3}$$

Where $\mathrm{k} = 1, 2, \ldots, \mathrm{N_{DOF,coh}}$. As previously mentioned, the hierarchical structure implemented in the FNM allows all the SEs to share information with each other, as they are part of the same laminate element. This feature of the FN laminate element is particularly convenient for the modelling of phenomena such as delamination migration [51].

4.1.4. Partitioning strategy and crack propagation in the FNM

In the previous sections, it has been described how, once a fracture initiation criterion is met, the floating nodes are directly used to explicitly represent the boundaries of a matrix crack. Furthermore, it has been illustrated how the tip of a crack can be closed through the introduction of internal nodes in the FN cohesive element formulation, obtaining the crack tip DOFs through interpolation from the DOFs of neighbouring nodes. In particular, the variables that dictate the partitioning strategy and assist crack propagation through the elements are the *element* and *edge status variables*. As a matter of fact, every element is associated a flag value which represents its status (”INTACT”, ”TRANSITION_ELEM”, ”REFINEMENT_ELEM”, ”CRACK_TIP_ELEM”, ”CRACK_WAKE_ELEM”, etc.). In a similar fashion, an edge status variable (”INTACT”, ”TRANSITION_EDGE”, ”REFINEMENT_EDGE”, ”CRACK_TIP_EDGE”, ”CRACK_WAKE_EDGE”, etc.) is associated to every edge. In addition to this, information about potential crack tip coordinates $\mathbf{x}_\mathrm{c}$ is also allocated to every edge. Such status variables depend on the location of a certain element/edge with respect to the discontinuity. As an example, let us consider the situation illustrated in Figure 4.7.



Edge #	Status	Crack coords
1	CRACK TIP EDGE	x_{C_I}
2	INTACT EDGE	

Edge #	Status	Crack coords
1	CRACK WAKE EDGE	x_{C_I}
2	CRACK TIP EDGE	$x_{C_{II}}$

Figure 4.7: Crack propagation example using the edge status variable approach: before propagation across element A (left); after propagation across element A (right)

In such circumstances, the crack tip is located at edge e_1, which is belonging to element A. As a result, the status variable of edge e_1 is ”CRACK_TIP_EDGE”. At the same time, edge e_1 is storing information about the planar coordinates of the crack $\mathbf{x}_{\mathrm{c}_\mathrm{I}}$. On the other hand, the status of element A is ”TRANSITION_ELEM”. In the FNM, a *transition element* allows partition, independently from the crack propagation direction, even though it is not cut by a crack yet. Due to such partition, the crack tip DOFs are not interpolated from adjacent nodes anymore, but are represented by floating nodes, so that a more accurate representation of the crack tip is obtained. To further improve the representation of the crack tip and to make direct use of the Virtual Crack Closure Technique (VCCT), the FNM can introduce a *refinement element* between the cracked element and the transition element [34].

Upon satisfaction of a certain fracture propagation criterion, element A updates its status variable from ”TRANSITION_ELEM” to ”CRACK_WAKE_ELEM”, changing its partition according to the crack angle. At the same time, the coordinates of the new crack tip $\mathbf{x}_{\mathrm{c}_\mathrm{II}}$ are calculated and stored to edge e_2. Edge e_1 updates its status variable from ”CRACK_TIP_EDGE” to ”CRACK_WAKE_EDGE”, as it is now cut by the crack. On the other hand, edge e_2 changes its status from ”INTACT” to ”CRACK_TIP_EDGE”, as it is hosting the crack tip. In a similar manner, element B becomes a transition element and partition itself accordingly.

If it is desired to prevent a discontinuity from propagating inside a certain element set, this can be achieved by using the ”tie” keyword in the input file. Upon preparation of the input file for the FNM part, the edges of such elements will be assigned a specific flag which gives them the label of *unbreakable edges*. The use of element and edge status variables provides an efficient way to carry out partitioning and crack propagation. As a matter of fact, every element is required to read only a limited amount of information to partition itself accordingly or allow propagation of the discontinuity [51].

4.2. Damage initiation and propagation criteria

In order to initiate and propagate damage in the laminate element, specification of fracture criteria is needed. These will serve to assess whether to carry out the partitioning and/or to degrade the stiffness of the laminate according to the type and extent of damage present.

4.2.1. Fibre damage

The damage initiation, as the name suggest, refers the the starting point of degradation of the structural response. Such degradation begins when the stresses/strains satisfy a specific criterion. In the current FNM implementation, a maximum stress criterion is used as damage initiation criterion for the fibres. Hence, considering both the tensile and compressive failure of the fibres, the failure index $\mathrm{f_f}$ is given by:

$$\begin{aligned} \mathrm{f_f} &= \frac{\sigma_{11}}{\mathrm{X_t}}; && \text{if } \sigma_{11} > 0 \\ &= \frac{\sigma_{11}}{\mathrm{X_c}}; && \text{if } \sigma_{11} < 0 \end{aligned} \tag{4.2.1.1}$$

Where σ_{11} is the stress in the fibre direction, $\mathrm{X_t}$ is the fibre tensile strength and $\mathrm{X_c}$ the fibre compressive strength. If the damage initiation criterion is satisfied ($\mathrm{f_f} \geq 1$), a bi-linear cohesive traction-separation law is used to model the damage propagation. The typical trend of such law is illustrated in Figure 4.8.

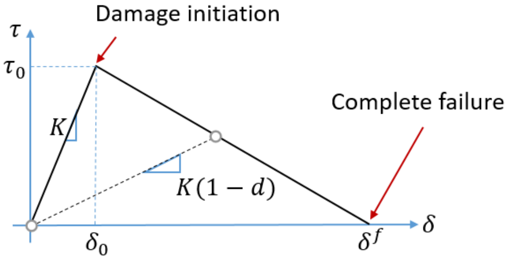


Figure 4.8: Traction-separation bi-linear cohesive law

Such law is characterized by a linear response up to damage initiation. Up to this point, there is no accumulation of damage, so that if unloading occurs the same stiffness curve is followed. After onset, a linear softening due to accumulation of damage is assumed up to complete failure. The softening is characterized by a damage parameter d, as pictured in Figure 4.8. In such case, the reduced stiffness path will be followed upon unloading. The damage parameter can assume values ranging from 0 (no damage) to 1 (complete failure), but it can never decrease.

Considering fibre damage, if failure onset is reached, the separation δ_0^{f} and traction τ_0^{f} at failure initiation are calculated as:

$$\begin{aligned} \delta_0 &= \frac{|\varepsilon_{11}|}{\mathrm{f_f}}\ell_{\mathrm{e}}; \\ \tau_0 &= \frac{|\sigma_{11}|}{\mathrm{f_f}}; \end{aligned} \tag{4.2.1.2}$$

Where ε_{11} and σ_{11} are the strain and the stress components in the fibre direction, while ℓ_{e} is the element characteristic length. In particular, the element characteristic length is defined as the length of the line segment parallel to the fibre angle, which passes through the element centroid and crosses two edges of the element. On the other hand, from LEFM it is known that the area under the traction-separation curve represents the energy absorbed by the material prior to complete failure, given the name of fracture toughness $\mathrm{G_c}$. As a result, since a bi-linear cohesive law is used, the fibre fracture toughness $\mathrm{G_{fc}}$ is given by the following:

$$\begin{aligned} \int_0^{\delta^{\mathrm{f}}} \sigma_{11}\,\mathrm{d}\delta &= \frac{1}{2}\mathrm{X_t}\delta^{\mathrm{f}} = \mathrm{G_{fc,T}} && \text{if } \sigma_{11} > 0; \\ &= \frac{1}{2}\mathrm{X_c}\delta^{\mathrm{f}} = \mathrm{G_{fc,C}} && \text{if } \sigma_{11} < 0 \end{aligned} \tag{4.2.1.3}$$

Where δ^{f} indicates the separation at complete failure for the fibre damage. Thus, separation at complete failure can be calculated from Equation 4.2.1.3, where all the other quantities are known. In particular, in the FNM implementation it is assumed that the fibre fracture toughness scales with respect to ply blocking:

$$\mathrm{G_{fc}^{n}} = \mathrm{n} \cdot \mathrm{G_{fc}^{1}} \tag{4.2.1.4}$$

Where n is the number of plies and $\mathrm{G_{fc}^{1}}$ is the fibre fracture toughness value for a single ply. The scaling is based on the empirical observations in [68], where ply blocking has been found to be linked to a higher energy dissipation per unit area of projected crack propagation, due to the occurrence of more fibre pull-out upon fracture [51].

The fibre damage parameter $\mathrm{d_f}$ at the onset of damage is calculated as follows:

$$\mathrm{d_f} = 1 - \frac{1}{\mathrm{f_f}} \tag{4.2.1.5}$$

Once the fibre damage parameter is determined, the elastic constants that are related to the fibres are degraded:

$$\begin{aligned} \mathrm{E_1^d} &= \mathrm{E_1}(1-\mathrm{d_f}); \\ \nu_{12}^{\mathrm{d}} &= \nu_{12}(1-\mathrm{d_f}); \\ \nu_{13}^{\mathrm{d}} &= \nu_{13}(1-\mathrm{d_f}); \end{aligned} \tag{4.2.1.6}$$

(4.2.1.7)

Where the d superscript is used to refer to the damaged elastic constants values. In the assessment of onset of damage for the fibres, it is assumed that fibre damage will also cause fibre-matrix debonding and matrix cracking. As a result, in the implemented damage algorithm, the onset of damage for the fibres corresponds to an onset of damage for the matrix. If fibre damage is already present in the element, the damage parameter is updated according to:

$$\mathrm{d_f} = \left(\frac{\delta^{\mathrm{f}}}{\delta_{\mathrm{eff}}}\right)\left(\frac{\delta_{\mathrm{eff}} - \delta_0}{\delta_{\mathrm{f}} - \delta_0}\right) \tag{4.2.1.8}$$

Where δ_{eff} is the effective separation for the fibre damage cohesive law.

4.2.2. Matrix/delamination damage

If there is no fibre damage in the element, a quadratic traction criterion is used to assess the damage onset for the matrix in the element. In the current FNM implementation, only tensile failure is considered and matrix cracks are assumed to be perpendicular to the plane of the laminate. As a result, in plane stress components are used for to assess damage onset in the matrix. Damage is initiated in the matrix when the failure index $\mathrm{f_m}$, to which a quadratic interaction function is associated, reaches a value of one:

$$\mathrm{f_m} = \left(\frac{\tau_\mathrm{n}}{\mathrm{Y_t}}\right)^2 + \left(\frac{\tau_\mathrm{t}}{\mathrm{S_t}}\right)^2 + \left(\frac{\tau_\ell}{\mathrm{S}_\ell}\right)^2 \tag{4.2.2.1}$$

Where τ_n, τ_t and τ_ℓ refer to the normal, transverse shear and longitudinal shear traction components acting on the matrix crack surface, respectively, while $\mathrm{Y_t}$, $\mathrm{S_t}$ and S_ℓ represent the tensile, transverse shear and longitudinal shear strength allowables for the matrix. Since matrix cracks are known to develop along the fiber direction and it is assumed that these are perpendicular to the plane of the laminate, the traction components are given by the followings:

$$\begin{aligned} \tau_\mathrm{n} &= \sigma_{22}; \\ \tau_\mathrm{t} &= \tau_{13}; \\ \tau_\ell &= \tau_{12}; \end{aligned} \tag{4.2.2.2}$$

Where σ_{22}, τ_{13} and τ_{13} refer to the stress components in the local material direction of the ply.

According to the partitioning procedure outlined in the previous sections, once matrix damage is initiated, a cohesive SE is formed to explicitly represent it but no stiffness degradation of the ply element is applied upon partition. For the new cohesive SE, the same damage initiation criterion is used. If damage onset is satisfied in the cohesive SE, the penalty stiffness values defining the cohesive constitutive law, $\mathrm{K_n}$, $\mathrm{K_t}$ and K_ℓ are degraded as follows:

$$\begin{aligned} \mathrm{K_n^d} &= \mathrm{K_n}(1-\mathrm{d_m}); \\ \mathrm{K_t^d} &= \mathrm{K_t}(1-\mathrm{d_m}); \\ \mathrm{K}_\ell^\mathrm{d} &= \mathrm{K}_\ell(1-\mathrm{d_m}); \qquad \text{with } \mathrm{d_m} = 1 - \frac{1}{\mathrm{f_m}} \end{aligned} \tag{4.2.2.3}$$

Where $\mathrm{d_m}$ is the matrix damage parameter. On the other hand, damage evolution for a matrix crack in the FNM uses an energy based power law mixed-mode criterion. The mixed-mode damage initiation and propagation can be visualized in Figure 4.9, where the s and n subscripts refer to the shear and normal contributions, respectively.

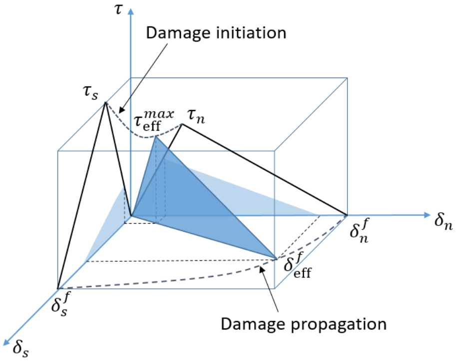


Figure 4.9: Mixed-mode bi-linear traction separation curve

This criterion determines propagation of a crack under mixed loading conditions assuming a power interaction function between the fracture toughness in the normal, transverse shear and longitudinal shear directions. Let δ_n, δ_t and δ_ℓ be the separation components and τ_n, τ_t and τ_ℓ the traction components in the cohesive element representing the matrix crack. To describe the damage evolution under mixed loading conditions, it is useful to introduce the effective separation δ_eff and effective traction τ_eff:

$$\begin{aligned} \delta_\mathrm{eff} &= \sqrt{\langle\delta_\mathrm{n}\rangle^2 + \delta_\mathrm{t}^2 + \delta_\ell^2}; \\ \tau_\mathrm{eff} &= \frac{\tau_\mathrm{n}\langle\delta_\mathrm{n}\rangle + \tau_\mathrm{t}\delta_\mathrm{t} + \tau_\ell\delta_\ell}{\delta_\mathrm{eff}} \\ &= \frac{\mathrm{G_n} + \mathrm{G_t} + \mathrm{G}_\ell}{\delta_\mathrm{eff}}; \qquad \text{with } \langle\delta_\mathrm{n}\rangle = \max(0, \delta_\mathrm{n}) \end{aligned} \tag{4.2.2.4}$$

Where $\mathrm{G_n}$, $\mathrm{G_t}$ and G_ℓ represent the normal tensile, transverse shear and longitudinal shear strain energy density. The bi-linear trend of the cohesive law implies:

$$\int_0^{\delta_\mathrm{eff}^\mathrm{f}} \tau_\mathrm{eff}\,\mathrm{d}\delta_\mathrm{eff} = \frac{1}{2}\tau_\mathrm{eff}^\mathrm{max}\delta_\mathrm{eff}^\mathrm{f} = \mathrm{G_{mc}} \tag{4.2.2.5}$$

Where $\tau_\mathrm{eff}^\mathrm{max}$ is the value of τ_eff at damage onset, similarly to the single mode case. The power law mixed-mode fracture toughness $\mathrm{G_{mc}}$ is then defined as:

$$\mathrm{G_{mc}} = \left[\left(\frac{\lambda_\mathrm{n}}{\mathrm{G_{nc}}}\right)^\alpha + \left(\frac{\lambda_\mathrm{t}}{\mathrm{G_{tc}}}\right)^\alpha + \left(\frac{\lambda_\ell}{\mathrm{G}_{\ell\mathrm{c}}}\right)^\alpha\right]^{-\frac{1}{\alpha}} \tag{4.2.2.6}$$

Where $\mathrm{G_{nc}}$, $\mathrm{G_{tc}}$ and $\mathrm{G}_{\ell\mathrm{c}}$ are the fracture toughness in the tensile normal, transverse shear and longitudinal shear directions, α is a material dependent exponent and where λ_n, λ_t and λ_ℓ are given by:

$$\lambda_\mathrm{n} = \frac{\mathrm{G_n}}{\mathrm{G_m}}; \qquad \lambda_\mathrm{t} = \frac{\mathrm{G_t}}{\mathrm{G_m}}; \qquad \lambda_\ell = \frac{\mathrm{G}_\ell}{\mathrm{G_m}}; \qquad \text{with } \mathrm{G_m} = \mathrm{G_n} + \mathrm{G_t} + \mathrm{G}_\ell \tag{4.2.2.7}$$

Upon damage propagation, the matrix damage parameter $\mathrm{d_m}$ is updated to further degrade the stiffness according to the following:

$$\mathrm{d_m} = \left(\frac{\delta^\mathrm{f}}{\delta_\mathrm{eff}}\right)\left(\frac{\delta_\mathrm{eff} - \delta_0}{\delta_\mathrm{f} - \delta_0}\right) \tag{4.2.2.8}$$

Where δ_0 and δ^f, which are the separation at failure onset and complete failure respectively, are calculated from the power law mixed-mode fracture toughness relation upon damage initiation.

In a similar fashion, delamination is modeled in the FNM with the same damage initiation and propagation criteria used for matrix cracks. The cohesive crack propagation criterion illustrated so far works in conjunction with the partitioning strategy described in Section 4.1.4. Hence, a crack is propagating from one edge to the other, allowing elements to share informations about the status of their common edges. Intact elements have the ability to partition themselves prior damage onset, in order to improve the representation of the crack tip.

4.2.3. Viscous regularization of damage

A viscous regularization has been implemented for both fibre and matrix crack/delamination damage parameters in order to help convergence. As a matter of fact, convergence difficulties are not uncommon when dealing with problems involving non-linear effects. Regarding damage propagation, an abrupt change of stiffness due to contact, or extensive propagation of damage might occur in a specific time increment. Considering the bi-linear cohesive law illustrated in Figure 4.8, convergence issues are caused by the sudden change in stiffness at the peak between the two linear parts of the curve. A common technique to overcome this problem in Abaqus is the use of *viscous regularization* for the damage variables used to degrade the elastic constants [59]. By specification of a viscous parameter μ_v, traction and separation values outside the cohesive law limits are allowed, in order to return a positive tangent stiffness matrix. The regularized damage d^v is related to the un-regularized damage d through the following law:

$$\frac{d(d^v)}{dt} = \frac{1}{\mu_v}(d - d^v) \tag{4.2.3.1}$$

To obtain the regularized damage parameter d^v to be used at the time increment $k+1$, the above equation in integrated with respect to time. Hence, the following is obtained:

$$d^v_{k+1} = \frac{\Delta t}{\mu_v + \Delta t} d_{k+1} + \frac{\mu_v}{\mu_v + \Delta t} d^v_k \qquad \text{with } \Delta t = t_{k+1} - t_k \tag{4.2.3.2}$$

Where d_{k+1} is the un-regularized damage parameter at increment $k+1$, calculated according to the cohesive softening law and d^v_k is the regularized damage parameter from the previous time increment. It is easy to verify that setting $\mu_v = 0$ in Equation 4.2.3.2 returns the un-regularized damage parameter. A rule of thumb to estimate an upper limit to be set for the viscous parameter μ_v is the following:

$$\mu_v < \frac{\delta_0}{\dot{u}} \tag{4.2.3.3}$$

Where δ_0 is the separation at damage onset according to the cohesive law of the material and $\dot{u}$ is the typical amplitude of nodal velocity in the mesh, which can be considered to be given by the loading rate. The use of viscous regularization with a small value of viscosity parameter μ_v has been observed to help convergence without compromising the results [59].

4.3. Combination of FNM and co-rotational formulation

In this section, the integration of the FN laminate element in the CR procedure outlined in Chapter 3 is proposed. In order to understand how it is possible to embed such element in the CR approach, it is worth to further consider the implementation scheme of the FN laminate element. This one is outlined in Figure 4.10.

As previously explained, the FN laminate element is defined such that an arbitrary number of FN ply and FN cohesive interface elements can be allocated as objects of such element. Every element is integrated separately and the individual contributions are assembled to build the stiffness matrix and internal force vector of the FN laminate element accordingly. Let us consider a single FN ply element which is part of the FN laminate element. Similarly to the FN laminate element, different types of elements can be allocated as objects of this element. If partition is triggered, integration will not be carried out on the standard 8-node solid element, but on the ”abstract” solid sub-bulks and on the cohesive SE representing the matrix crack. The cohesive FN element representing the interface works in a similar fashion. If the FN cohesive element is partitioned, integration will be performed on the top and/or bottom cohesive SEs, rather than on the ”abstract” intact cohesive element.

In either case, integration is performed on elements which are completely equivalent to standard brick/wedge solid and cohesive elements, highlighted in red in Figure 4.10. As a result, it suffices to apply the CR procedure to these elements to include geometric non-linear effects in the FN laminate element. Integration of the solid wedge element in the CR procedure proved to be straightforward, as this element belongs to the same element class to which the hexahedral element is belonging. The computational scheme used for the CR-wedge element does not change and follows the procedure illustrated in Figure A.7, where the only difference is in the number of nodes of the element.

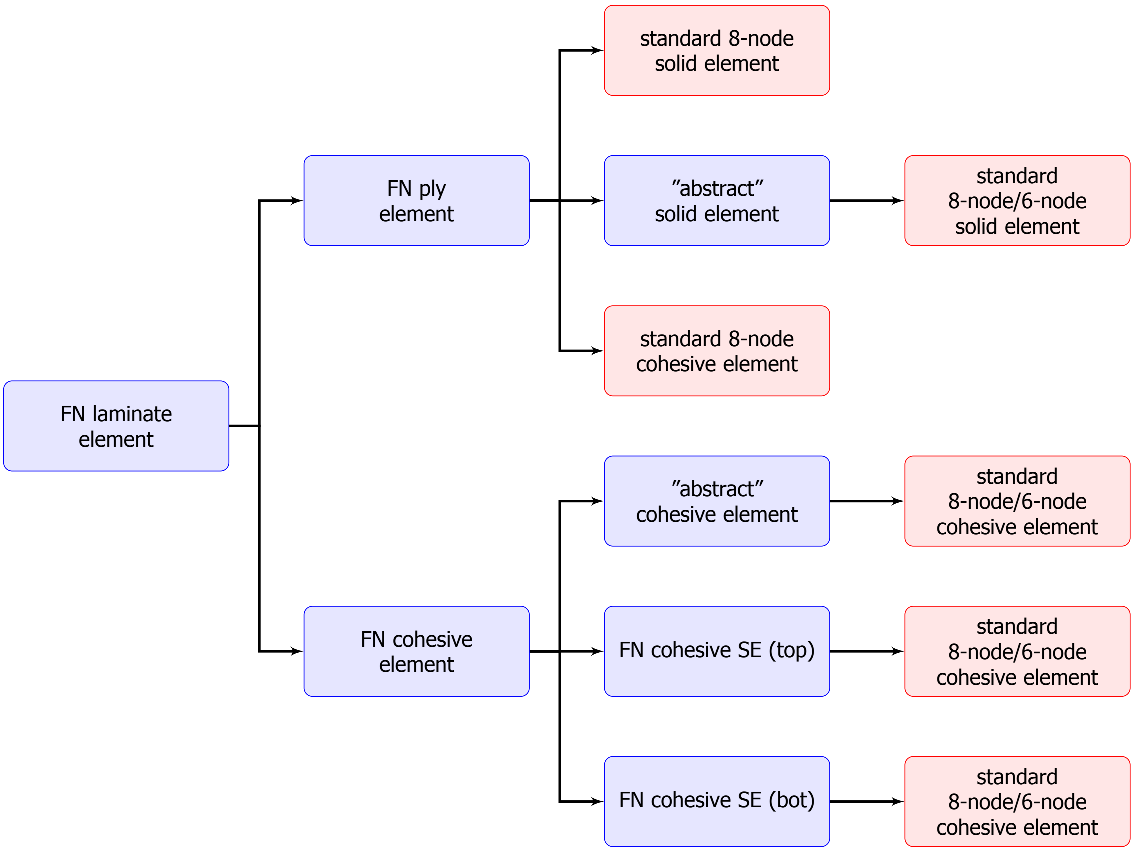


Figure 4.10: Hierarchical structure of the FN laminate element

On the other hand, cohesive elements have a substantial different formulation, so that the same CR approach developed for solid elements cannot be applied. However, a particularly cheap procedure to include geometric non-linearities in these elements is to refer all the element calculations to the *last converged mid-plane configuration*, rather than to the undeformed mid-plane configuration. The two are pictured in Figure 4.11 for convenience.

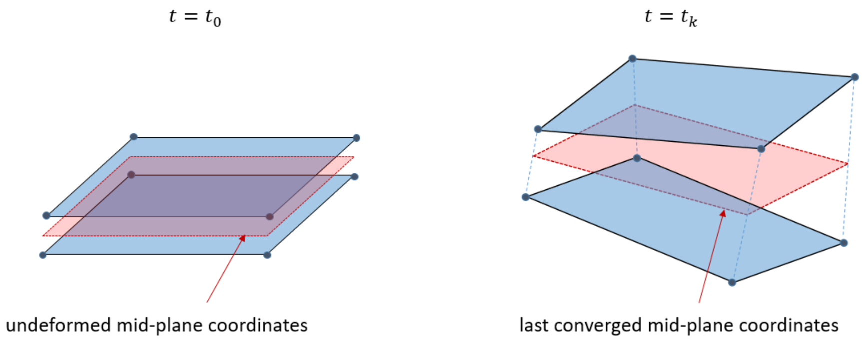


Figure 4.11: Visual illustration of undeformed (left) and last converged (right) mid-plane configuration for the cohesive element

As a matter of fact, it is worth reminding that the cohesive element is sharing its top and bottom surface with the solid FN elements, which already include geometric non-linearities via the CR approach. The last converged nodal coordinates of the cohesive element for the k-th iteration are calculated as:

$$\mathbf{x}_{\mathrm{k}} = \mathbf{x} + \mathbf{u}_{\mathrm{k}} - \Delta\mathbf{u}_{\mathrm{k}} \tag{4.3.0.1}$$

Where the nodal incremental displacements $\Delta \mathbf{u}_{\mathrm{k}}$, which are passed in to the UEL by Abaqus, are removed from the total displacements to obtain the last converged configuration. Finally, the last converged mid-plane coordinates $\mathbf{x}_{\mathrm{mid,k}}$ are given by:

$$\mathbf{x}_{\mathrm{mid,k}} = \frac{1}{2}(\mathbf{x}_{\mathrm{k}}^{\mathrm{bot}} + \mathbf{x}_{\mathrm{k}}^{\mathrm{top}}) \tag{4.3.0.2}$$

The suitability of such approach for cohesive elements has been determined through testing of the single cohesive element UEL, in comparison with the Abaqus standard cohesive element. A simple model in which the cohesive seam, attached to solid elements, is subjected to excessive out of plane deflection has been built in Abaqus. While the results from the linear formulation were found not to be in agreement with the Abaqus reference, the formulation using last converged mid-plane coordinates for element calculations proved to be equivalent to the Abaqus non-linear cohesive formulation. As a matter of fact, it can be observed that the two contour plots of the displacement magnitude corresponds to each other, as shown in Figure 4.12

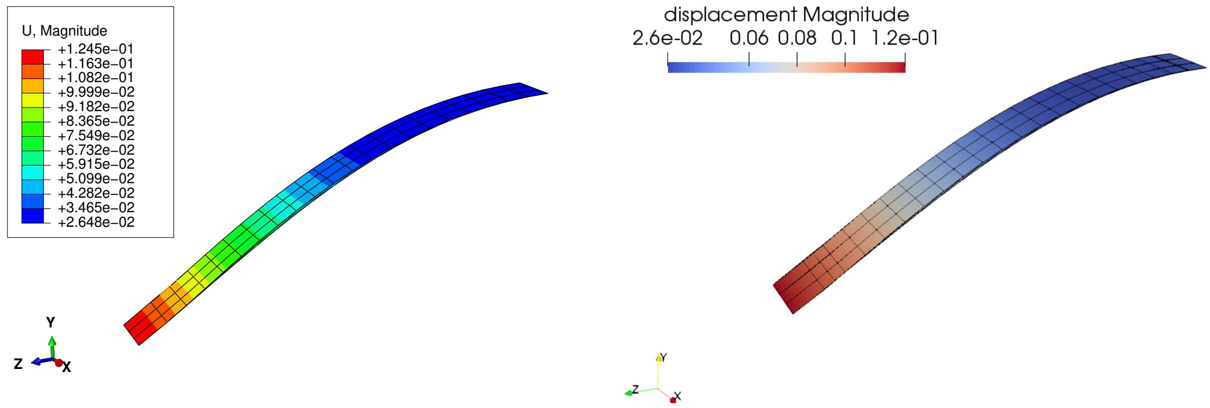


Figure 4.12: Displacement contour plot comparison between the Abaqus reference model and the cohesive UEL

Most importantly, the two models were found to be characterized by the same delamination growth pattern throughout the analysis, as illustrated in Figure 4.13, proving that the implemented cohesive formulation is able to correctly represent damage evolution in the model. Figure 4.13 compares the stiffness degradation variable (SDEG) in Abaqus to the delamination damage (dd) in ParaView, showing very close agreement between the two.

As a result, a new element has been formed, which is given the name of CR-FN laminate element. Such FN element includes the geometric non-linearities in its solid SEs via the CR approach presented in Chapter 3, while it uses the above described modified cohesive formulation for the FN interface SEs. With such element, the modelling of crushing and low-velocity impacts phenomena has been addressed and it is presented in the next section.

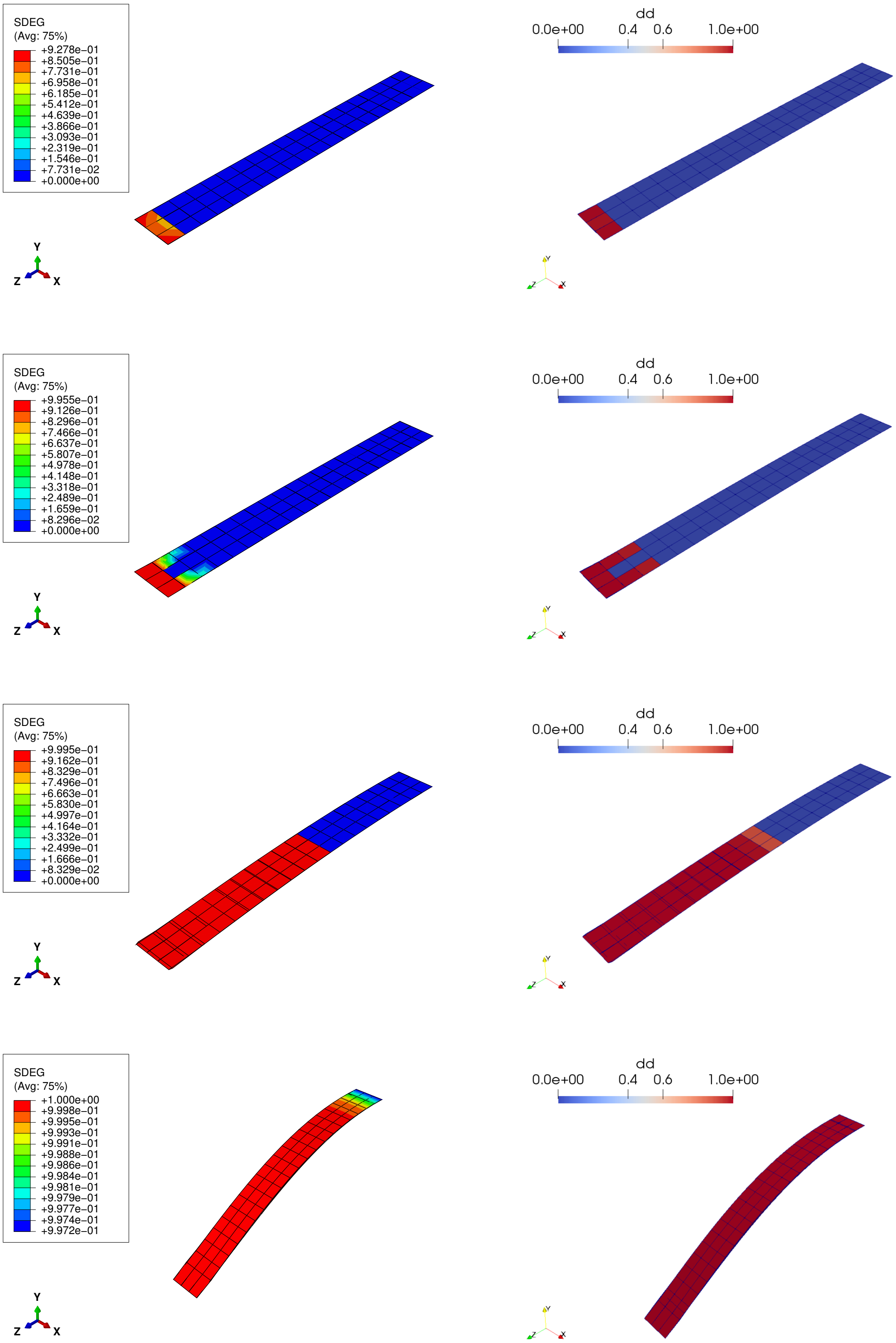


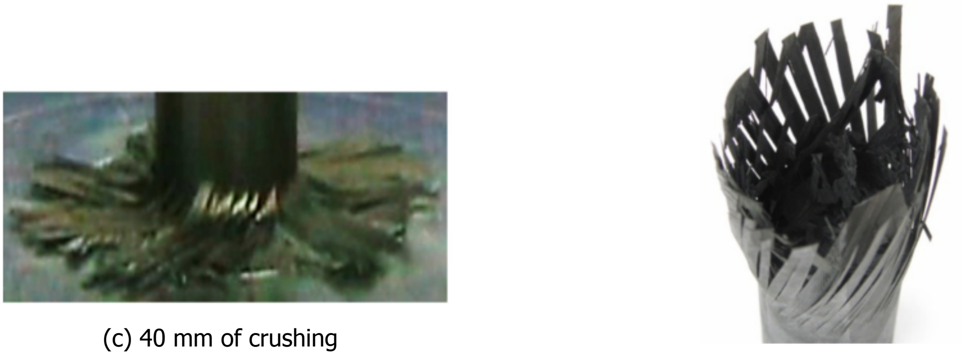
Figure 4.13: Delamination growth comparison between the Abaqus reference model and the cohesive UEL

4.3.1. Towards the modelling of axial crushing and low-velocity impact phenomena

Structural crashworthiness is known to be a crucial requirement in many engineering sectors, especially in the aerospace and automotive industry [69]. Initially, metals have been used to manufacture energy absorption tubular components. As a matter of fact, metals were found to be convenient due to their progressive plastic folding behavior [70]. Nevertheless, in the last decades Fiber Reinforced Polymers (FRP) became particularly attractive for applications where energy absorption and weight saving are the main design drivers. However, these materials are known to experience very little or no plasticity at all. On the contrary, when subjected to axial crushing, they undergo several fracture mechanisms in order to absorb energy. In particular, experimental findings in [71] suggest that most of the energy absorption is obtained through the Mode I splitting of the plies, bending of the fronds, Mode II driven delamination, fibre fracture and friction at and within the fronds. The typical damage pattern observed is illustrated in Figure 4.14.

(a) 5 mm of crushing

(b) 20 mm of crushing



(c) 40 mm of crushing

(d) Final deformation

Figure 4.14: Axial crushing of a $[+15/\text{-}15/+15/0_3/+15/\text{-}15/+15]$ circular CFRP tube [1]

In order to test the element performance in capturing such a phenomenon a reduced test model has been developed. Assuming an axis-symmetric damage pattern, a small strip in the composite tube is modelled to capture the frond bending behavior. The strip has length $\ell = 5\mathrm{mm}$, width $b = 1\mathrm{mm}$ and layup $[0/90]$. The assumed material properties are given in Table 4.1. In particular, the penalty stiffness used for the cohesive interface elements has been estimated according to [72]:

$$K \approx 50\frac{E_3}{h} \tag{4.3.1.1}$$

Where E_3 is the out of plane elastic modulus and h the total thickness of the laminate.

The reference model built in Abaqus is shown in Figure 4.15. In the two-plies model, the translational displacements are constrained at the left end. At the right end, a velocity load $V_1 = -1\mathrm{mm/s}$ is applied to the outer nodes of the 0 ply, which are also constrained from moving in the y direction. The part has been meshed with 5 elements along its length direction and 1 element in the width direction. The velocity load is applied in a total time of $t_{tot} = 2\mathrm{s}$, with a fixed time increment $\Delta t = 2 \cdot 10^{-3}\mathrm{s}$. Imposition of a compressive load on the 0 ply causes the laminate to deflect due to the difference in

Property	**Value**	**Units**
Longitudinal modulus E_1	120.2	[GPa]
Transverse modulus E_2, E_3	8.724	[GPa]
Shear modulus G_{12}, G_{13}	4.556	[GPa]
Shear modulus G_{23}	3.980	[GPa]
Poisson ratio v_{12}, v_{13}	0.32	[–]
Poisson ratio v_{23}	0.436	[–]
Longitudinal tensile strength X_t	2060	[MPa]
Longitudinal compressive strength X_c	1400	[MPa]
Transverse tensile strength X_t	41.50	[MPa]
Transverse compressive strength X_t	185	[MPa]
Shear strength S_t, S_ℓ	90	[MPa]
Single ply fibre fracture toughness $G^1_{fc,T}$, $G^1_{fc,C}$	112.7	$[kJ/m^2]$
Mode I matrix/interfacial fracture toughness G_{nc}	0.212	$[kJ/m^2]$
Mode II matrix/interfacial fracture toughness G_{tc}, G_{lc}	0.774	$[kJ/m^2]$
Power law exponent α	1	[–]
Penalty stiffness K_n, K_t, K_ℓ	$2 \cdot 10^6$	[MPa/mm]
Single ply thickness t_{ply}	0.125	[mm]

Table 4.1: Material properties used for the composite model

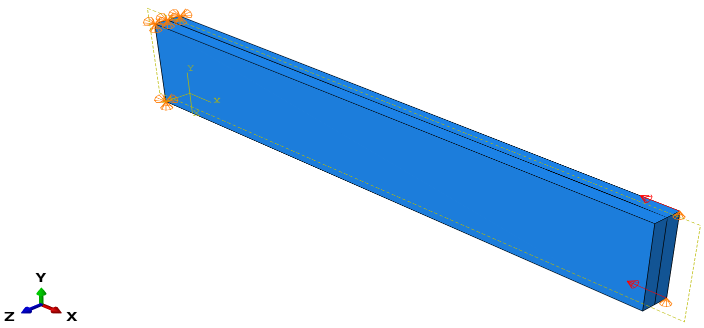
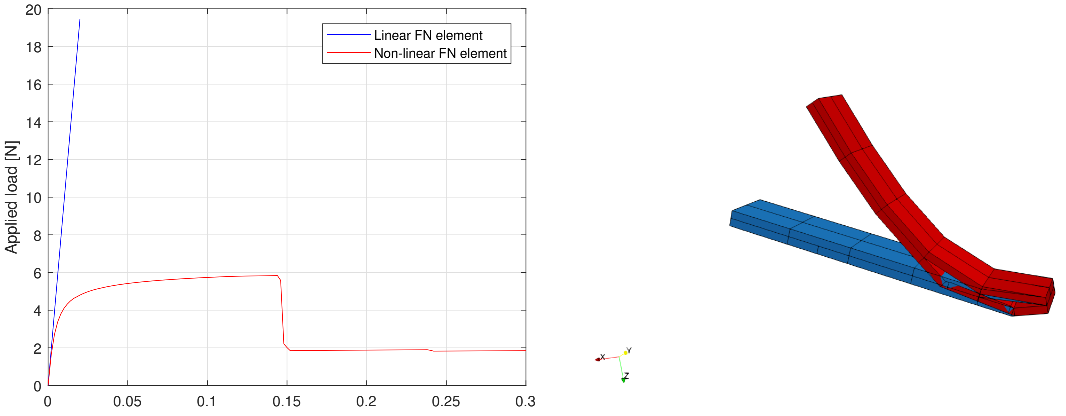


Figure 4.15: Reference 2-plies model to simulate frond bending

modulus between the plies in the longitudinal direction. A comparison with the linear version of the code is given in Figure 4.16. As it can be observed, the load-displacement curve is far from linear in the non-linear version of the FN laminate element. The softening observed in the non-linear model is due to the complete failure of the interface and occurrence of matrix cracks, formed at the left end at an early stage compared to the linear counterpart. These features can be explicitly visualized in ParaView, as shown in Figure 4.17



(a) Comparison between the load-displacement curves

(b) Comparison between deformed shape

Figure 4.16: Comparison between linear and geometric non-linear FN laminate element output results

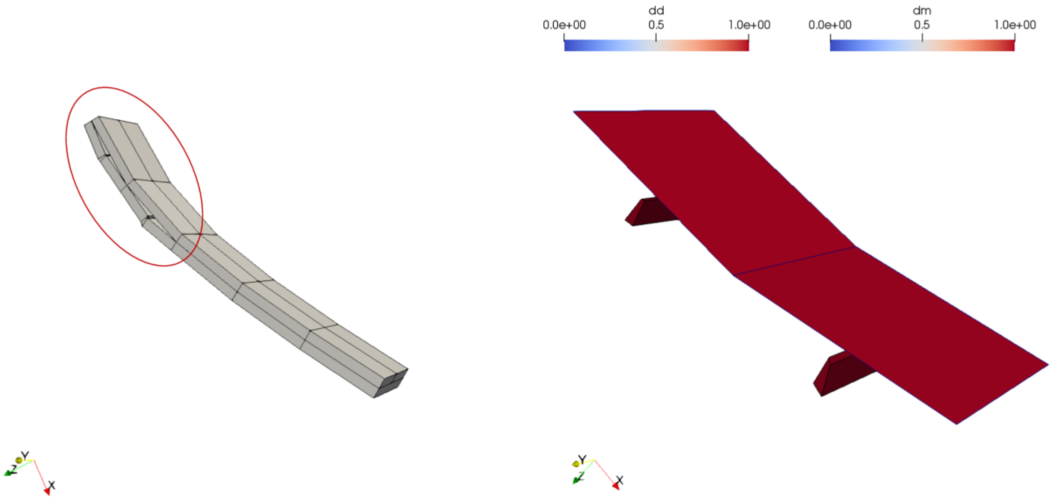


Figure 4.17: Visualization of delamination damage (dd) and matrix damage (dm) in ParaView

The development of transverse matrix cracks in the 90 ply due to shear is qualitatively a faithful representation of the phenomenon. On the other hand, little delamination damage and matrix damage has been observed in the linear model. In particular, the latter showed an excessive amount of fibre damage, present also in the non-linear model but in a smaller amount. By increasing the mesh seed in the longitudinal direction a higher matrix crack density has been obtained in the 90 ply.

As an example of the element potential in explicitly representing matrix cracks and delamination damage in impact phenomena, a small test model, illustrated in Figure 4.18, has been developed. This one consists in a squared plate, where a velocity load is incrementally imposed to its center nodes. The plate has simply supported boundary conditions applied at its vertical edges, while the horizontal edges are free, and has stacking sequence $[0/90]_s$. The material properties reported in Table 4.1 have been used for this model, where the penalty stiffness of the cohesive elements has been modified according to Equation 4.3.1.1. The damage propagation in the model can be visualized in Figure 4.19. The damage pattern developed in a symmetric manner, as it would be expected for a symmetric laminate.

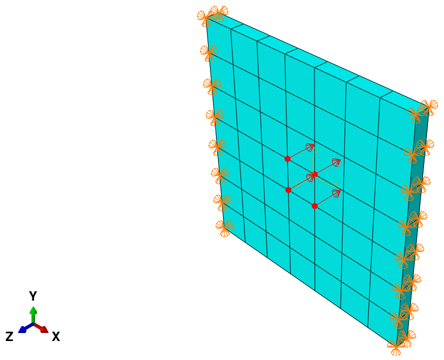
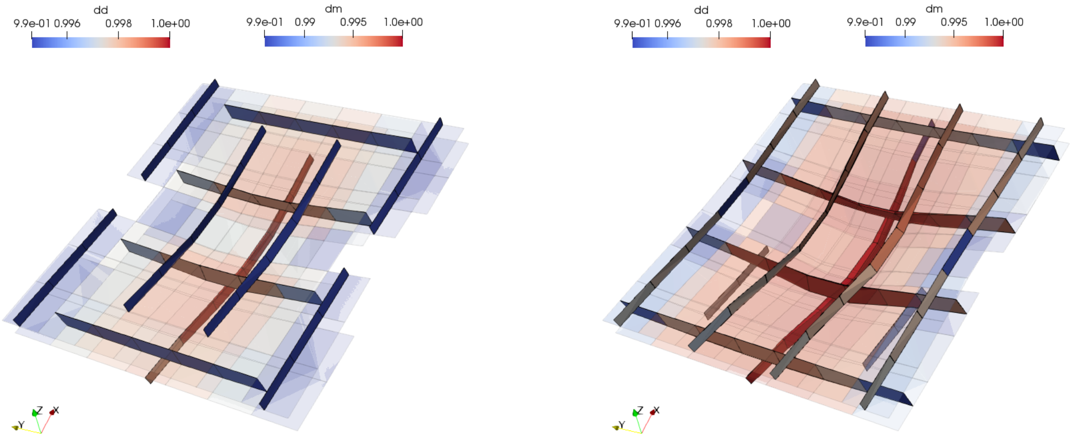
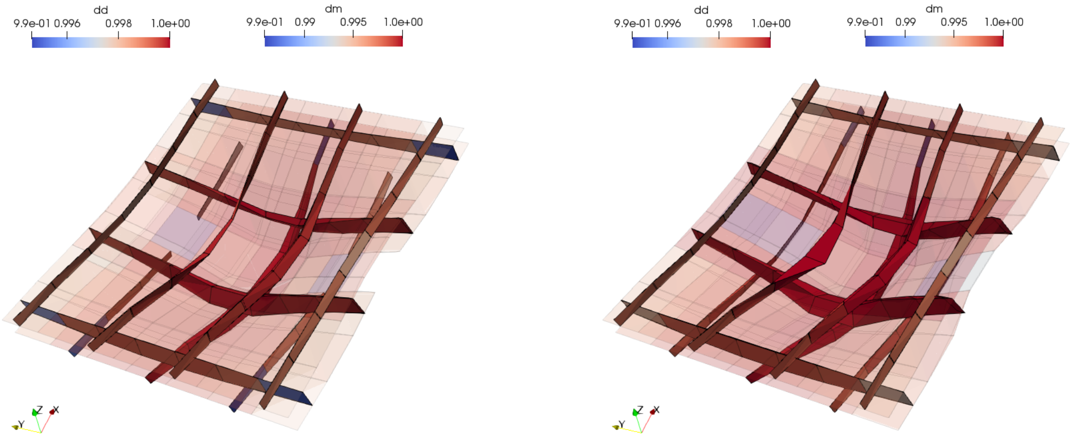


Figure 4.18: Test model to simulate impact loading on a composite laminate



(a) Matrix/delamination damage at $t = 0.25t_{tot}$

(b) Matrix/delamination damage at $t = 0.5t_{tot}$



(c) Matrix/delamination damage at $t = 0.75t_{tot}$

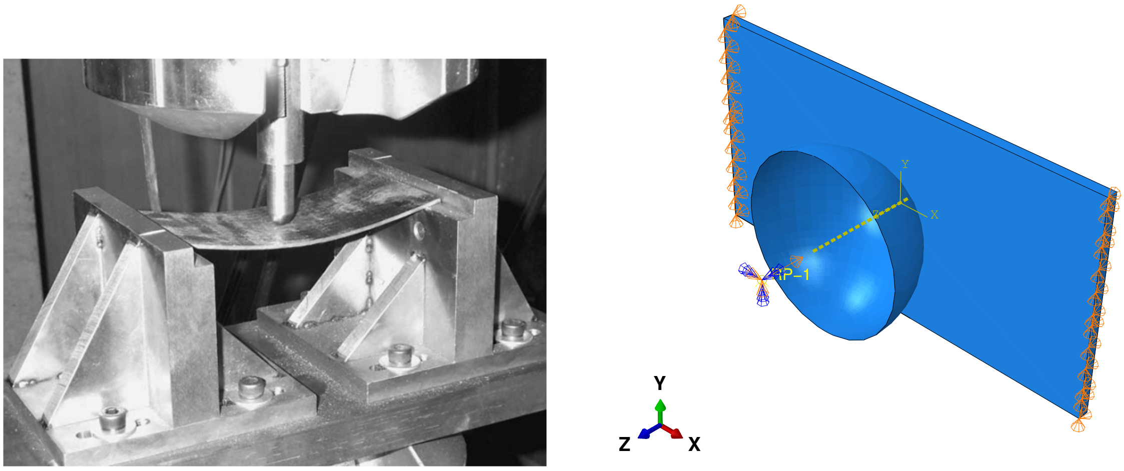
(d) Matrix/delamination damage at $t = t_{tot}$

Figure 4.19: Visualization of matrix (dm) and delamination (dd) damage with the FNM

4.3.2. Quasi-static crushing of a cross-ply laminate

This model refers to the work of Huang et al. [73] and has been chosen as benchmark due to the explicit mentioning of geometric non-linear effects. In their research, two CFRP rectangular plates are tested in quasi-static crushing up to failure. The laminates, both with a total number of 16 plies, have stacking sequence $[0/90]_{4s}$ and $[-45/45]_{4s}$, and are named *cross-ply* and *angle-ply*, respectively. In the model, the 0 fiber direction is defined as the one in the longitudinal direction of the plates. The crush test fixture is such that it offers simply supported boundary conditions on the two shorter edges, leaving the two longer edges free. The experimental setup is illustrated in Figure 4.20a. In the experiment, the hemispherical steel impactor loads the composite plate up to ultimate failure. The recorded load displacement curve shows a significant drop in the force due to sudden failure of the laminates.

In an attempt to reproduce the original 16-plies model, high computational efforts have been observed. These are believed to be linked to the not fully consistent tangent stiffness matrix in the CR formulation and to the occurrence of multiple damage softening mechanisms. As a result, a reduced model of the cross-ply laminate with 4 plies only has been developed in order to draw a qualitative comparison. The main interest is in capturing the geometric non-linear trend, with and without failure, of the load-displacement curve and to compare the obtained damage pattern to the one observed in the experiments. The reduced 4-plies model built in Abaqus is illustrated in Figure 4.20b.



(a) Experimental setup [73]

(b) Reduced 4-plies reference model

Figure 4.20: Quasi-static crushing of a cross-ply laminate

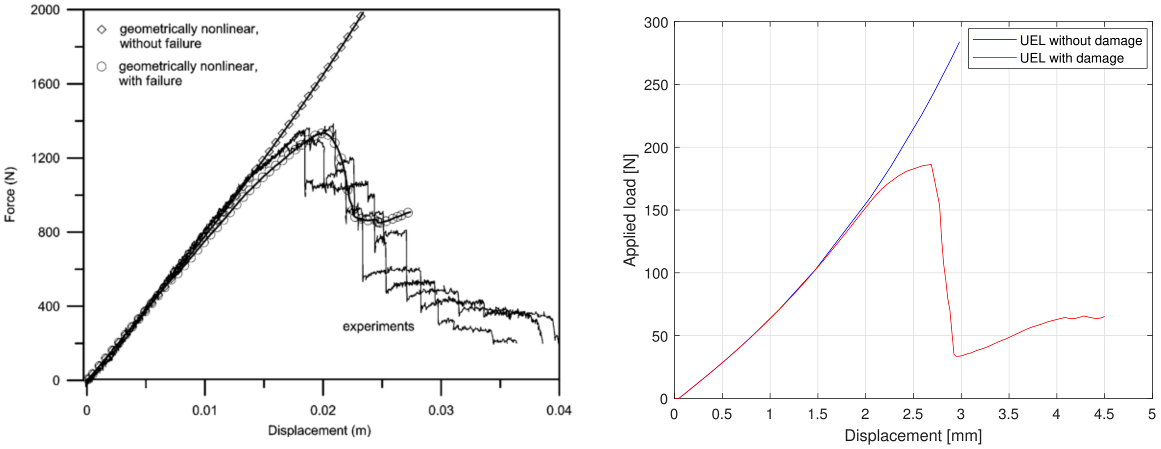
In the reference model, the laminate has lenght $\ell = 20\mathrm{mm}$, width $\mathrm{b} = 10\mathrm{mm}$ and stacking sequence $[0/90/90/0]$, with a single ply thickness $\mathrm{t}_{\mathrm{ply}} = 0.125\mathrm{mm}$. The reference model has been meshed with 800 elements, with only 1 element through the thickness as requested by the Python pre-processing script. As only part of the material properties needed for the model were given in [73], the material properties reported in Table 4.1 have been used.

In the model, the part is constrained from moving in the z and y directions at the two shorter edges. The steel impactor has been modeled as a 3D discrete rigid part, and associated to the reference point RP-1 using a rigid body constraint. A displacement load of $4.5\mathrm{mm}$ in the negative z direction has been applied on the reference point RP-1. It is worth to remind here that, since Abaqus does not recognize the UEL topology, only node-based surface definitions could have been used. As a result, it has not been possible to define the indentor as an analytical rigid body, which is often used in contact problems.

Surface-to-surface contact has been specified, with the master surface being the impactor and the slave surface a set of preselected contact nodes at the center of the laminate. A node-to-surface discretization algorithm has been used as it was possible to specify only node-based surfaces for the contact definition. On the other hand, a finite sliding formulation has been used, since this one is the one suggested in the Abaqus User's Manual for geometrically non-linear analyses [59]. In the contact property definition, a tangential behavior with friction coefficient $\mu_\mathrm{t} = 0.23$, based on [74], and a hard contact pressure-overclosure for the normal behavior has been specified, with a penalty contact algorithm. As a matter of fact, the latter results in a less stringent enforcement of contact and it has been observed to facilitate convergence. A viscous parameter $\mu_\mathrm{v} = 1 \cdot 10^{-5}$ has been used for both the fiber and matrix/delamination damage. Convergence has been obtained without specification of contact damping. However, contact damping can be specified in the Interaction Properties module to help convergence, through definition of a damping coefficient μ_d. As a rule of thumb, an estimation for the damping coefficient can be obtained using information from the contact pressures and nodal velocities in the model before inclusion of damping, considering the following relation [59]:

$$\mathrm{f_{vd}} = \mu_\mathrm{d} \mathrm{A} \dot{\mathrm{u}}_\mathrm{rel}^\mathrm{el} \tag{4.3.2.1}$$

Where $\mathrm{f_{vd}}$ are the damping forces, A the nodal area and $\dot{\mathrm{u}}_\mathrm{rel}^\mathrm{el}$ the relative motion between the two surfaces, which can be obtained examining the message ($\mathrm{.msg}$) file. A comparison between the results reported in the reference and ones obtained in the 4-plies model is given in Figure 4.21b.

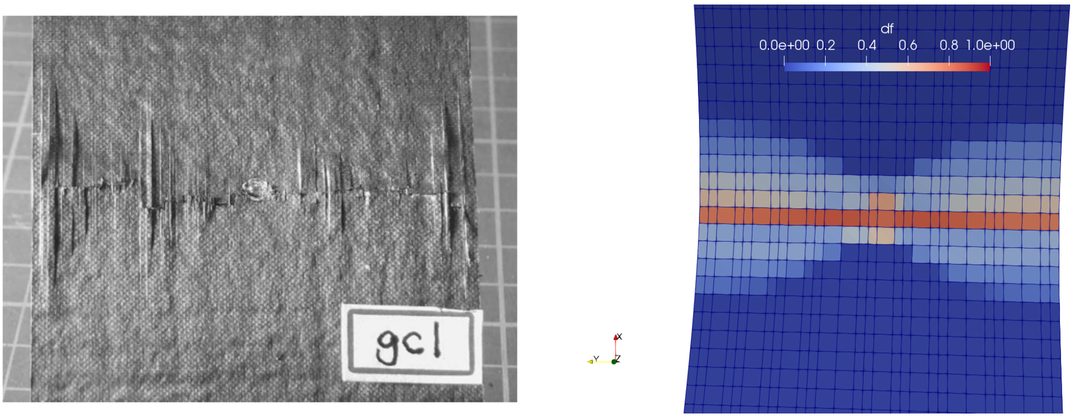


(a) Reference load-displacement curves for the cross-ply laminate [73]

(b) Obtained load-displacement curves for the reduced cross-ply laminate model

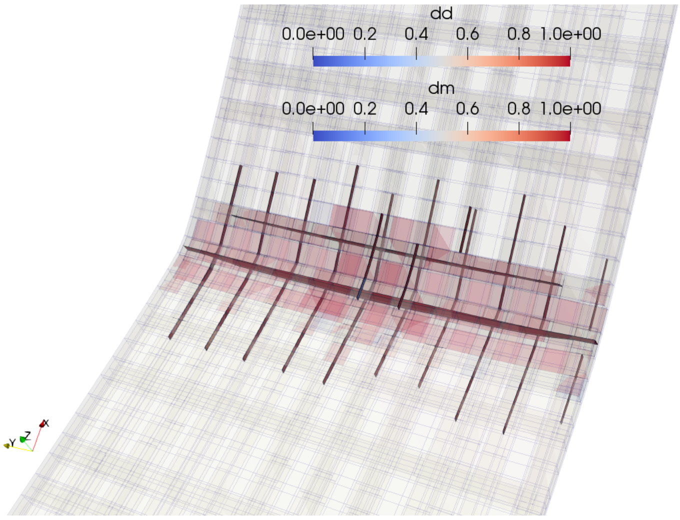
Figure 4.21: Qualitative comparison of the load-displacement curve

From the comparison with the reference it can be observed that a similar geometric non-linear trend is captured by the UEL. However, a much higher loss of stiffness is observed in the reduced model, due to an abrupt evolution of damage. At the same time, good agreement with the experiment is observed in Figure 4.22, which illustrates the damage pattern on the front of the laminate. As a matter of fact, a similar butterfly-like shape for the fibre damage has been obtained, as reported in Figure 4.22b, with the formation of longitudinal matrix cracks as pictured in Figure 4.22c. Good agreement with the experiment can also be observed considering the back view of the damaged laminate, given in Figure 4.23.



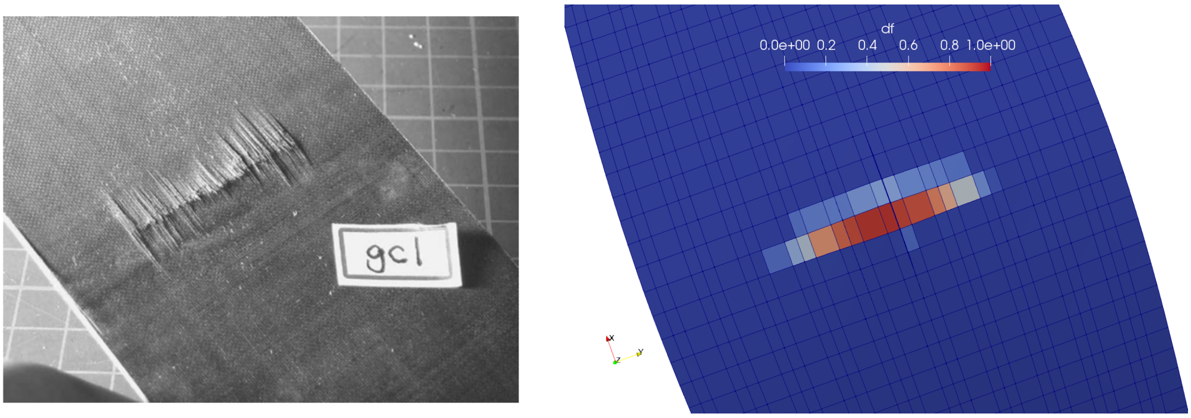
(a) Front view of the damaged cross-ply laminate [73]

(b) Front view of the fibre damage (df) contour plot



(c) Front view of matrix damage (dm) and delamination damage (dd)

Figure 4.22: Qualitative comparison of the damage pattern on the front of the cross-ply laminate



(a) Back view of the damaged cross-ply laminate [73]

(b) Back view of the fibre damage (df) contour plot

Figure 4.23: Qualitative comparison of the fiber damage pattern on the back of the cross-ply laminate

5

Conclusions

5.1. Conclusion

Fibre-reinforced composites have become increasingly attractive for many engineering applications in the last decades. A very interesting aspect of these materials is that their mechanical properties can be tailored for optimum strength and stiffness by controlling the orientation of the fibers. At the same time, they are characterized by high strength properties, strong corrosion resistance, improved damage tolerance and can lead to considerable weight and cost reduction when compared to their metallic counterparts. However, the accurate modelling of damage in composites is still an active research topic, as their progressive failure involves the interaction of various intra- and inter-laminar damage mechanisms which often lead to complex fracture paths. In particular, the modelling becomes even more challenging in phenomena where geometric non-linear effects are not negligible, such as a crush or impacts.

Concerning the modelling of damage, enriched methods such as XFEM and PNM proved to have several advantages over standard FEM fracture modelling techniques, as they allow to model a discontinuity inside a FE domain. However, these also present some drawbacks, among which the inability to explicitly represent discontinuities, their difficulty in modelling complex cracking scenarios and the intrinsic error in the mapping of a straight crack to the parent element domain [34]. The novel FNM, implemented by Chen et al. [34], offers an element formulation which proved to overcome these issues. The peculiarity of this approach is that its formulation leads to the exact same solution as that obtained through a local remeshing explicitly representing the discontinuities. Furthermore, the FNM proved to be particularly suited for the modelling of damage in composite structures.

The objective that has been set for this thesis was to extend the capabilities of the FN laminate element by including geometric non-linear effects in the element formulation, in order to address complex fracture phenomena such as crushes or impacts. To this end, following the literature study, the relatively new CR approach was considered to be a very suitable option. As a matter of fact, the CR approach has been found to provide a convenient way to incorporate geometric non-linear effects in a FE, while retaining element independence in terms of implementation. This formulation essentially consists in a front end filtering operation, lying between the assembler/solver and the existing finite element library. In order to fulfill the final objective of the thesis, the problem has been tackled into several steps.

An eight-node linear hexahedral element has first been implemented and validated in comparison to the standard C3D8 Abaqus elements through development of test models. As the final scope of this project was to potentially model dynamic problems, the extension of the element implementation to dynamic procedures has also been investigated. To facilitate the verification, a simplified implementation reduced to a Newmark scheme using trapezoidal rule has been developed. The procedure has been tested through a simple dynamic benchmark problem, which yielded results in agreement with a proposed analytical solution implemented in Matlab, as reported in Appendix A.

In a second stage, the CR procedure has been implemented and applied on the eight-node linear solid element. Even though the present implementation does not result in a fully consistent tangent stiffness matrix, the procedure has been successfully validated with three popular geometric non-linear problems reported by Sze et al. [56]. In two of the three models that were reproduced, it has been necessary to use a selective integration scheme in order to achieve the same degree of accuracy provided by the reference solutions and avoid the shear locking effect. In regard to convergence, the CR user element has been found to require less increments in two models out of three when compared to the C3D8 element. However, the use of selective integration has been observed to negatively affect convergence and increase the number of iterations needed to complete the analysis. In the third model, the CR element has been found to require a particularly high number of iterations. This is believed to be linked to the high aspect ratio of the elements in the mesh, as in the reference S4R elements were used [56]. Despite the good accuracy of the results, the CR user element has been observed to be characterized by a high computational time in comparison to standard Abaqus elements.

Finally, the extension of the FN laminate element to geometric non-linear analyses has been addressed. As integration in the FN laminate element is reduced to standard Gauss integration on the SEs domains, it has been judged theoretically sufficient to embed its ”standard” solid and cohesive SEs in the CR formulation to incorporate geometric non-linear effects. Although the CR procedure has been successfully applied to include geometric non-linear effects in the solid hexahedral/wedge SEs, it has been verified that the same approach could have not been applied to the cohesive SEs, due to substantial differences in terms of implementation. However, upon testing of the single cohesive UEL, a convenient approach has been found to include geometric non-linearities in the cohesive formulation. The approach consists in referring the cohesive element calculations to the last converged mid-plane coordinates. As a matter of fact, the cohesive top and bottom surface nodes are shared with the CR solid SEs, which already includes geometric non-linear effects. Upon comparison with an Abaqus reference cohesive model, this approach was found to provide the same results and to correctly capture the damage initiation/propagation. During testing of the so formed CR-FN laminate element, this one has been found to be characterized by a particularly high computational time. This is believed to be associated to the not fully consistent tangent stiffness matrix in the present implementation and the multiple, simultaneous damage softening occurring in the element. To this regard, a viscous regularization has been implemented for the damage parameters, which was observed to ease convergence. The CR-FN laminate element has been used to reproduce some geometric non-linear test models. Substantial differences have been observed in the comparison with the linear FN laminate element and a qualitative comparison suggests that the element has an interesting potential in addressing geometric non-linear fracture problems.

5.2. Recommendations

Further work is certainly needed for this research to reach the full potential of the method. In particular, the following recommendations are made:

- The implemented CR formulation is currently using a SVD algorithm to obtain the orthogonal polar factor of the deformation gradient of the element. Although SVD is a well established and validated algorithm, it is also known to be computationally expensive. This might be linked to the high computational time required by the CR user element to complete the analysis. As a result, other algorithms to obtain the orthogonal polar factor should be considered.

- The accuracy of the results obtained with the CR formulation proves that the internal force vector of the element is calculated correctly. However, the present CR formulation does not result in a fully consistent tangent stiffness matrix. This is believed to be associated to the high computational efforts observed for the element. Furthermore, it is known that an inconsistent tangent stiffness matrix may provide inaccurate results in buckling problems. Therefore, further work is needed to implement a fully consistent tangent stiffness matrix.

- In the implemented implicit dynamic procedure, the control parameters have been set to values equivalent to a Newmark scheme with trapezoidal rule, to facilitate the verification of the solution with the proposed analytical solution. However, this set of values is often not the most appropriate one, depending on the amount of energy dissipation involved in the physical phenomenon to be modeled. At the same time, the damping matrix contribution has been neglected. As a result, further improvements of the implementation are needed if it is desired to address dynamic problems.

- Concerning the high computational efforts required by the CR-FN laminate element, it is believed that the simultaneous occurrence of different types of damage in the element makes it difficult for the solver to find equilibrium. As a result, a more in depth investigation regarding the influence of damage softening on the rate of convergence of the analysis is required to improve the element performance.

A

Appendix

A.1. Extension of element validity for dynamic implicit analyses

In this section, the extension of the linear brick element for dynamic implicit analyses is investigated. The section starts with an introduction to the dynamic problem. Next, the main differences between *explicit* and *implicit* time integration schemes are outlined to the reader. In conclusion, a detailed overview of the implementation of this procedure in the Abaqus UEL subroutine is given at the end of the section, to end with a verification example.

A.1.1. The dynamic problem

The dynamic implicit procedure is typically used in Abaqus/Standard to investigate the non-linear dynamic response of the structure. In Section 2.3, a static analysis procedure was considered. This one implies that all the loads are *static* or *quasi-static*, which means they are either constant over time or they do vary with such a slow rate that there is no substantial variation in the structural response. If these assumptions are not valid anymore, a dynamic analysis should be performed.

For dynamic problems in FEM, a set of discretized equilibrium equations that include the inertial and damping terms is solved, which can be written as:

$$\mathbf{M}\ddot{\mathbf{u}}(\mathrm{t}) + \mathbf{C}\dot{\mathbf{u}}(\mathrm{t}) + \mathbf{K}\mathbf{u}(\mathrm{t}) = \mathbf{f}^{\mathrm{ext}}(\mathrm{t}) \tag{A.1.1.1}$$

Where $\mathbf{M}$ is the mass matrix, $\mathbf{C}$ the damping matrix, $\mathbf{K}$ the stiffness matrix and $\mathbf{f}^{\mathrm{ext}}$ the external forces applied to the system. The presence of the damping term is accounting for the presence of different kinds of damping forces, which resist the dynamic motion. Even though it is difficult to quantify these forces, it is reasonable to assume that these are velocity-dependent. On the other hand, the damping matrix $\mathbf{C}$ does not depend on geometric and material properties of the system. Among the existing methods to construct such matrix, the most used is the Rayleigh damping. According to this method, the damping matrix is given as a combination of the mass and stiffness matrices of the system, multiplied by suitable constants α and β which are problem dependent:

$$\mathbf{C} = \alpha\mathbf{M} + \beta\mathbf{K} \tag{A.1.1.2}$$

To facilitate the verification, the damping contribution has been neglected in the present implementation so it will not be considered hereafter. The solution to a dynamic problem can be obtained either by modal analysis method or by direct integration method [75, 76]. However, when there is a high degree of non-linearity in the system, only the use of direct integration methods is suitable. In order to obtain a set of algebraic equations, a discrete approximation in time is assumed for the solution, so that:

$$\mathbf{u}(\mathrm{t_n}) \approx \mathbf{u}_\mathrm{n}; \qquad \dot{\mathbf{u}}(\mathrm{t_n}) \approx \dot{\mathbf{u}}_\mathrm{n}; \qquad \ddot{\mathbf{u}}(\mathrm{t_n}) \approx \ddot{\mathbf{u}}_\mathrm{n} \tag{A.1.1.3}$$

As a result, the equilibrium equation at each discrete time can be written in the residual form as:

$$\begin{aligned}\mathbf{\Delta R}_{\mathrm{n}} &= \mathbf{f}_{\mathrm{n}}^{\mathrm{ext}} - \mathbf{M}\ddot{\mathbf{u}}_{\mathrm{n}} - \mathbf{K}\mathbf{u}_{\mathrm{n}} \\ &= \mathbf{f}_{\mathrm{n}}^{\mathrm{ext}} - \mathbf{M}\ddot{\mathbf{u}}_{\mathrm{n}} - \mathbf{f}_{\mathrm{n}}^{\mathrm{int}}\end{aligned} \tag{A.1.1.4}$$

A.1.2. Explicit/Implicit time integration schemes

In order to solve the system given in Equation A.1.1.4, two categories of existing approaches can be found in literature, namely *explicit* and *implicit* methods. Abaqus/Explicit is using the former. With this method, if the mass matrix is diagonal and the damping is considered to be only mass-proportional, the entire system of equations can be decoupled and solved without resorting to matrix factorization. Hence, choosing the nodal accelerations as single unknown:

$$\ddot{\mathbf{u}}_{\mathrm{n}} = \mathbf{M}^{-1}(\mathbf{f}_{\mathrm{n}}^{\mathrm{ext}} - \mathbf{f}_{\mathrm{n}}^{\mathrm{int}}) \tag{A.1.2.1}$$

The nodal displacements and velocities are then obtained using the central difference time integration scheme [22]. As a result, the solver does not have to iterate to obtain the solution. This feature makes the explicit solver computationally efficient. However, a severe drawback, of this method is that a mass-proportional damping affects mainly the low-frequency response of the structure, which is usually the one of main interest for design purposes. Another disadvantage of the explicit method is that the central difference scheme is *conditionally stable*, which means that the solution becomes unstable and rapidly diverges when the time increment is too large. In particular, the following limitation applies to the time increment $\Delta\mathrm{t}$ to be used in an explicit analysis:

$$\Delta\mathrm{t} \leq \Delta\mathrm{t}_{\mathrm{crit}} = \frac{\ell_{\mathrm{e}}}{\mathrm{c}} \tag{A.1.2.2}$$

Where ℓ_{e} is the characteristic element length and c is the stress wave speed of the material. As a result, the time increment should be smaller than the time that a stress wave takes to propagate through an element.

On the other hand, the implicit procedure solves a system of non-linear equations, dependent on quantities at time t_{n} as well as on quantities at time $\mathrm{t}_{\mathrm{n+1}}$. Such feature makes this method computationally more expensive compared to the explicit method. However, the aforementioned limitation on the time increment size does not apply to the implicit method, which is *unconditionally stable*. Furthermore, this one proved to be more suitable to investigate the low-frequency response in a system [76].

In order to solve this system of non-linear equations, the problem could be formally re-formulated as a first order system and solved through the application of multi-step methods. However, in structural dynamics it is preferred to use single-step methods, directly applied to the second-order system of equations. The most popular is the *Newmark method* [22]. This one relates the quantities at time $\mathrm{t}_{\mathrm{n+1}}$ to the quantities at time t_{n} through the following:

$$\mathbf{u}_{\mathrm{n+1}} = \mathbf{u}_{\mathrm{n}} + \Delta\mathrm{t}\dot{\mathbf{u}}_{\mathrm{n}} + \left(\frac{1}{2} - \beta\right)\Delta\mathrm{t}^2\ddot{\mathbf{u}}_{\mathrm{n}} + \beta\Delta\mathrm{t}^2\ddot{\mathbf{u}}_{\mathrm{n+1}} \tag{A.1.2.3}$$

$$\dot{\mathbf{u}}_{\mathrm{n+1}} = \dot{\mathbf{u}}_{\mathrm{n}} + (1-\gamma)\Delta\mathrm{t}\ddot{\mathbf{u}}_{\mathrm{n}} + \gamma\Delta\mathrm{t}\ddot{\mathbf{u}}_{\mathrm{n+1}} \tag{A.1.2.4}$$

Where β and γ are parameters selected to control accuracy, stability and frequency dissipation in the system. By substituting Equation A.1.2.3 in Equation A.1.1.4, it is possible to solve the system by choosing the displacement at the $\mathrm{t}_{\mathrm{n+1}}$ as primary unknown. As a result the residual equilibrium equation at time $\mathrm{t}_{\mathrm{n+1}}$ is given in the following form:

$$\mathbf{\Delta R}_{\mathrm{n+1}} = \mathbf{f}_{\mathrm{n+1}}^{\mathrm{ext}} - \mathbf{M}\frac{1}{\beta\Delta\mathrm{t}^2}(\mathbf{u}_{\mathrm{n+1}} - \bar{\mathbf{u}}_{\mathrm{n}}) - \mathbf{f}_{\mathrm{n+1}}^{\mathrm{int}} \tag{A.1.2.5}$$

Where $\bar{\mathbf{u}}_{\mathrm{n}}$ indicates all the contributions of the quantities at time t_{n}. The Abaqus/Standard implicit solver is using the Hilber-Hughes-Taylor time integration scheme (HHT). This one is an extension of the Newmark method that improves the numerical dissipation and has second order accuracy, not achievable with the regular Newmark method [77]. The HHT scheme can also be found in literature as α*-method*, as a new control parameter α is introduced modifying the residual equilibrium equation according to the following:

$$\mathbf{\Delta R}_{n+1} = (1+\alpha)\mathbf{f}^{ext}_{n+1} - \alpha\mathbf{f}^{ext}_{n} - \mathbf{M}\frac{1}{\beta\Delta t^2}(\mathbf{u}_{n+1} - \bar{\mathbf{u}}_n) - \mathbf{f}^{int}_{n+1} \tag{A.1.2.6}$$

The system is solved by iterative Newton method. The previously introduced control parameters are then related to the single control parameter α through the following:

$$\beta = \frac{1}{4}(1-\alpha^2) \tag{A.1.2.7}$$

$$\gamma = \frac{1}{2} - \alpha \tag{A.1.2.8}$$

The three control parameters can be tuned simultaneously to achieve a certain degree of energy dissipation in the system, even though by default β and γ are expressed in terms of α. In particular, three categories of dynamic applications are considered in the Abaqus manual [59]:

- *Transient fidelity applications*: this category is used when there is particular interest in the vibrational response of the structure. Therefore the control parameters should be tuned to have minimum energy dissipation.
- *Moderate dissipation applications*: this category includes a wide range of problems which can have either a monotonic or non-monotonic structural response. Since the vibrational behavior of the structure is not of main interest in these applications, some energy dissipation is inserted to improve convergence.
- *Quasi-static applications*: in this category are falling all those problems in which the main interest is the static response of the structure, but may exhibit unstable behavior. As a result, this can be characterized by relatively higher energy dissipation to improve the convergence.

A.1.3. Implementation of the dynamic procedure

For the dynamic implicit procedure, the mass matrix $\mathbf{M}$ of the element needs to be calculated, which is defined by Equation 2.2.1.14. However, in the FE formulation, the mass matrix is calculated resorting to numerical integration, so that:

$$\mathbf{M} = \sum_{k=1}^{n_Q} N_I(\boldsymbol{\xi}_k) N_J(\boldsymbol{\xi}_k) \det\mathbf{J}(\boldsymbol{\xi}_k) w(\boldsymbol{\xi}_k) \tag{A.1.3.1}$$

Where n_Q is the number of integration points and $\boldsymbol{\xi}$ are the natural coordinates of the integration points. It is worth mentioning that reduced quadrature should not be used to calculate the mass matrix, as spurious results can be obtained due to loss in rank of the mass matrix [78]. Furthermore, diagonalization of the mass matrix can significantly reduce the computational efforts, while not compromising the results in an irreversible way. In order to diagonalize the mass matrix, the three most used methods that can be found in literature are the following:

- *Row sum method*: $\tilde{M}_{ii} = \sum_j M_{ij}$

  The diagonal mass matrix is constructed by setting the diagonal entry equal to the sum of the terms of the whole row.
- *Diagonal scaling*: $\tilde{M}_{ii} = cM_{ii} \quad c = \frac{M_{tot}}{tr(M)}$

  The diagonal entry of the newly constructed matrix is set equal to the diagonal entry of the original mass matrix, scaled by a factor c.
- *Evaluation of* $\mathbf{M}$ *at the physical nodes*

  The mass matrix is evaluated by performing numerical integration at the physical nodes, yielding a diagonal matrix. However, this method often leads to loss of positive definiteness, so it is not preferred.

Among these, the row sum method has been used in the present implementation. In order to extend the validity of the element to dynamic implicit procedures, the LFLAGS array has been used. This one is an array provided to the UEL by Abaqus, which contains the flag parameters that define the solution procedure to be used and the element calculations to be performed. As a result, the dynamic procedure has been coded by extending the original implementation with additional *if statements*. The implemented *if*/*else if* structure is illustrated in Figure A.8. For completeness, an explanation is given here below:

- *LFLAGS(1)=1 or LFLAGS(1)=2*

  This flag corresponds to the static procedure. The same calculations as in the previous section are carried out at the element level. The jacobian matrix $\mathbf{K}$ and internal force vector $\mathbf{f}^{\mathrm{int}}$ have to be returned to Abaqus in the AMATRX and RHS variables respectively.

- *LFLAGS(1)=11 or LFLAGS(1)=12*

  This flag indicates the dynamic implicit procedure with half-incremental residual tolerance or fixed time increments respectively.

- *LFLAGS(3)=1*

  Flag for dynamic implicit procedure characterized by normal time increment. In this case, both the jacobian matrix and the internal force vector need to be specified.

  $$\mathrm{RHS} = \mathbf{M}\ddot{\mathbf{u}}_{\mathrm{t+\Delta t}} + \alpha \mathbf{f}_{\mathrm{st,t}} - (1+\alpha)\mathbf{f}_{\mathrm{st,t+\Delta t}} \tag{A.1.3.2}$$

  $$\mathrm{AMATRX} = (1+\alpha)\mathbf{K} + (1+\alpha)\mathbf{C} + \mathbf{M}\Big(\frac{1}{\beta \Delta \mathrm{t}^2}\Big) \tag{A.1.3.3}$$

- *LFLAGS(3)=5*

  Flag for dynamic implicit procedure with half-incremental residual calculation, which has to be defined for automatic time incrementation. In this case only the internal force vector has to be returned to Abaqus.

  $$\mathrm{RHS} = \mathbf{M}\ddot{\mathbf{u}}_{\mathrm{t}+\frac{\Delta \mathrm{t}}{2}} - (1-\alpha)\mathbf{f}_{\mathrm{st,t}+\frac{\Delta \mathrm{t}}{2}} + \frac{\alpha}{2}(\mathbf{f}_{\mathrm{st,t}} + \mathbf{f}_{\mathrm{st,t_0}}) \tag{A.1.3.4}$$

  Where $\mathbf{f}_{\mathrm{st,t_0}}$ is the static residual at the beginning of the previous increment.

- *LFLAGS(3)=4*

  Flag corresponding to velocity jump calculation, which is performed at the beginning of the dynamic step or when there is a velocity jump due to change in contact. These calculations are necessary to obtain velocities that are consistent with the constraints applied to the model. In such circumstances, the jacobian is equal to the mass matrix and the internal force vector is set to zero.

  $$\mathrm{AMATRX} = \mathbf{M} \tag{A.1.3.5}$$

  $$\mathrm{RHS} = 0 \tag{A.1.3.6}$$

- *LFLAGS(3)=6*

  Flag corresponding to the initial acceleration calculation. This is similar to the velocity jump calculation, as it is performed at the beginning of the dynamic step or after a change in contact. In order to satisfy equilibrium, the jacobian should be equal to the mass matrix and the internal force vector should contain only static and damping terms.

  $$\mathrm{AMATRX} = \mathbf{M} \tag{A.1.3.7}$$

  $$\mathrm{RHS} = \mathbf{f}_{\mathrm{st}} \tag{A.1.3.8}$$

A.1.4. Stress wave propagation in a beam

In order to verify the implemented dynamic implicit procedure, a stress wave propagation problem in a beam was chosen as benchmark. The model is illustrated in Figure A.1. In the problem, a beam with length $\ell = 1\mathrm{m}$ and a cross-sectional area $A = 1 \cdot 10^{-4}\mathrm{m}^2$ is subjected to a compressive axial load $F = 10\mathrm{N}$ at its left end, while it is clamped at its right end. The beam is assumed to be at rest at time $t = 0$. The beam is made out of an isotropic material with elastic modulus $E = 70\mathrm{GPa}$ and density $\rho = 2700\mathrm{Kg}/\mathrm{m}^3$. A fixed incrementation scheme with increment $\Delta t = 2 \cdot 10^{-6}\mathrm{s}$ is used, while the total time of the analysis is $t_{tot} = 5 \cdot 10^{-4}\mathrm{s}$. The part has been meshed with 50 elements along its longitudinal direction and with 1 element in the thickness direction.

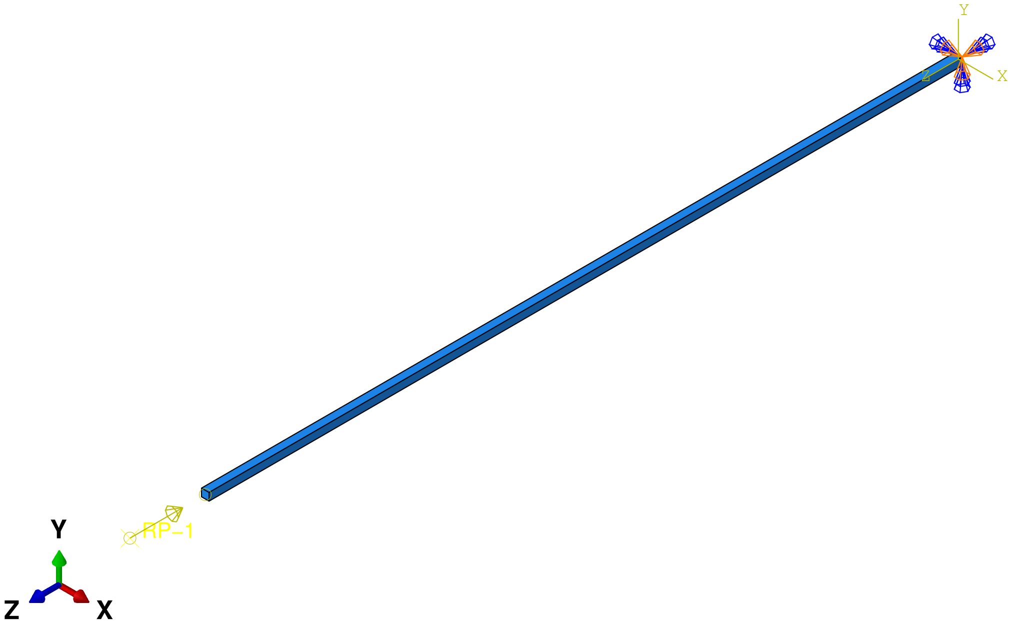


Figure A.1: Reference model for the stress wave propagation in a beam

To verify the solution, a Newmark time integration scheme was implemented in Matlab with trapezoidal rule. This scheme corresponds to the following values of control parameters:

$$\alpha = 0 \tag{A.1.4.1}$$

$$\beta = 0.25 \tag{A.1.4.2}$$

$$\gamma = 0.5 \tag{A.1.4.3}$$

By substitution of Equation A.1.2.3 and A.1.2.4 in Equation A.1.1.4, a system of Equations having the accelerations at time t_{n+1} as single unknown is obtained:

$$\mathbf{A}_{n+1}\ddot{\mathbf{u}}_{n+1} = \mathbf{b}_{n+1} \tag{A.1.4.4}$$

Where the matrix $\mathbf{A}$ indicates the terms multiplying the acceleration $\ddot{\mathbf{u}}_{n+1}$ while $\mathbf{b}_{n+1}$ indicates the other terms. The implemented scheme to obtain the solution at every time increment is shown in Figure A.2.

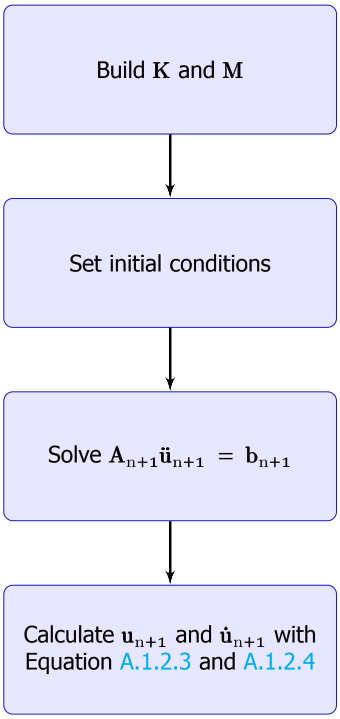


Figure A.2: Implemented Newmark scheme with trapezoidal rule in Matlab

The strain in the elements has been then calculated through the following:

$$\boldsymbol{\varepsilon}_n = \frac{\mathbf{u}_{n+1} - \mathbf{u}_n}{\ell_e} \tag{A.1.4.5}$$

Where ℓ_e is the characteristic element length.

The output stress contour plot at time $t = 0.5t_{tot}$ is shown in Figure A.3. In the figure it is possible to see the stress wave as it moves to the left after being reflected at the fixed end. On the other hand, the comparison between the implemented approach in Matlab and the UEL are shown in Figure A.4. As it can be observe, there is good agreement between the two.

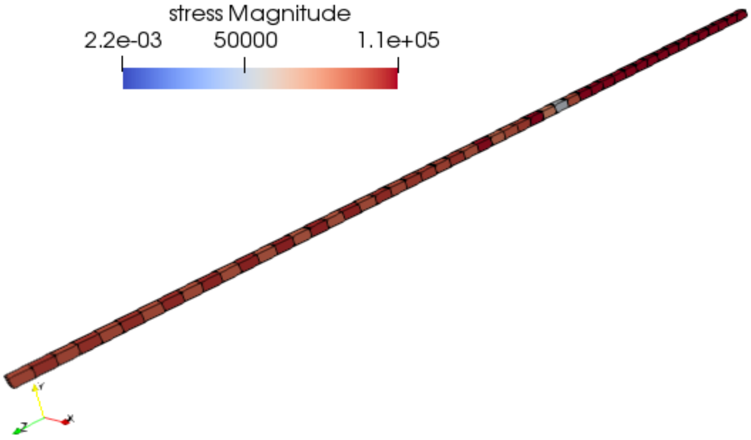


Figure A.3: Stress wave propagation in the beam contour plot

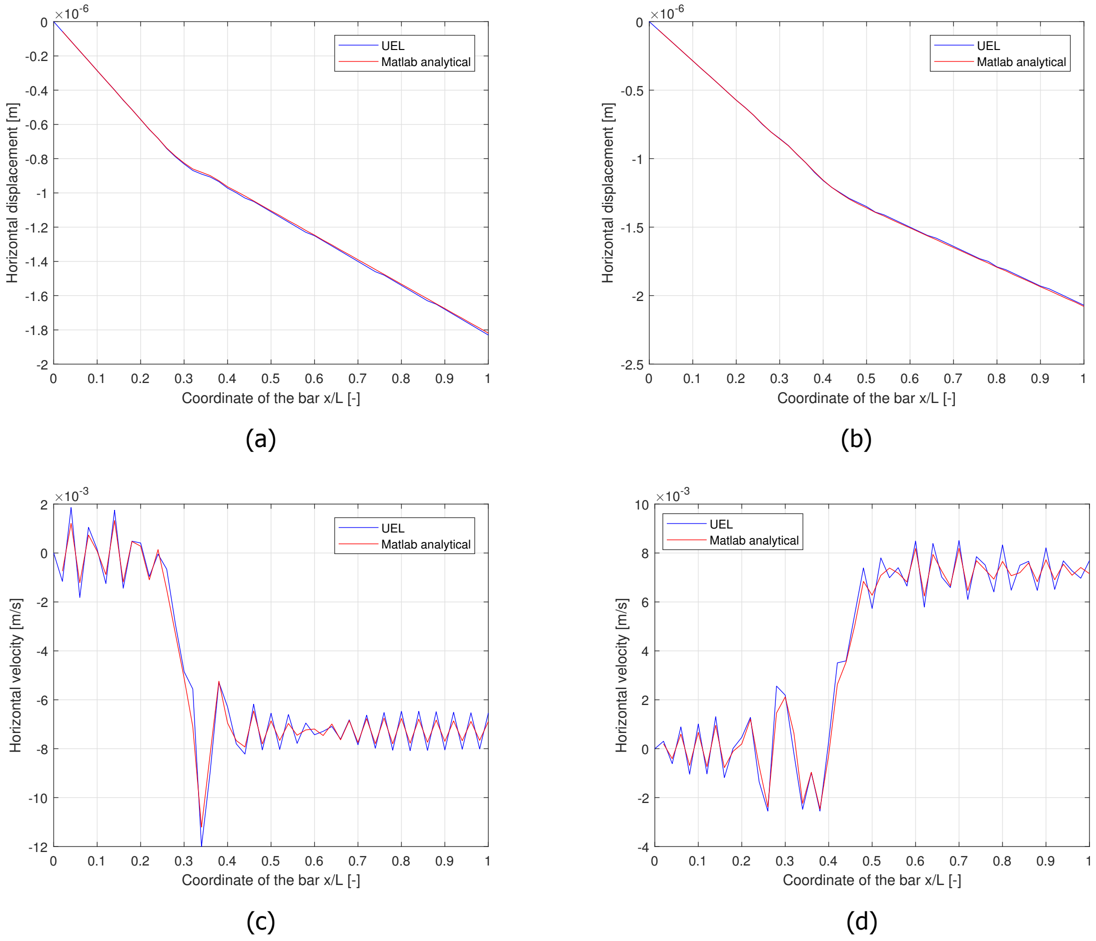


Figure A.4: At the top: horizontal displacement at time $t = 0.5t_{tot}$ (a) and at time $t = t_{tot}$ (b); At the bottom: horizontal velocity at time $t = 0.5t_{tot}$ (c) and at time $t = t_{tot}$ (d)

A.2. The 8-node linear cohesive element

In this section the implementation of the 8-node linear cohesive element is presented. The element topology is illustrated in Figure A.5.

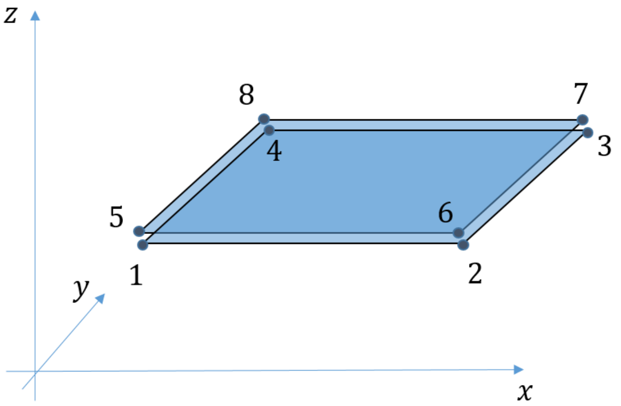


Figure A.5: Topology of an 8-node cohesive element

The element position vector and displacement vector are given by:

$$\begin{aligned} \mathbf{x}_{\mathrm{CE}} &= [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_8]^{\mathrm{T}} = [\mathbf{x}^{\mathrm{bot}}, \mathbf{x}^{\mathrm{top}}]^{\mathrm{T}} \\ \mathbf{u}_{\mathrm{CE}} &= [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_8]^{\mathrm{T}} = [\mathbf{x}^{\mathrm{bot}}, \mathbf{x}^{\mathrm{top}}]^{\mathrm{T}} \end{aligned} \tag{A.2.0.1}$$

In cohesive elements, element calculations are referred to the mid-plane coordinates. The mid-plane coordinates are calculated as:

$$\begin{aligned} \mathbf{x}_{\mathrm{mid}} &= \frac{1}{2}\left(\mathbf{x}^{\mathrm{bot}} + \mathbf{x}^{\mathrm{top}}\right) \\ &= [\mathbf{x}_1^{\mathrm{m}}, \mathbf{x}_2^{\mathrm{m}}, \mathbf{x}_3^{\mathrm{m}}, \mathbf{x}_4^{\mathrm{m}}]^{\mathrm{T}} \end{aligned} \tag{A.2.0.2}$$

Using the mid-plane coordinates, the normal $\mathbf{e}_{\mathrm{n}}$, tangent transverse $\mathbf{e}_{\mathrm{t}}$ and tangent longitudinal $\mathbf{e}_{\ell}$ vectors to the mid-plane surface of the cohesive element can be obtained as follows:

$$\begin{aligned} \mathbf{e}_{\ell} &= \frac{\mathbf{x}_2^{\mathrm{m}} - \mathbf{x}_1^{\mathrm{m}}}{|\mathbf{x}_2^{\mathrm{m}} - \mathbf{x}_1^{\mathrm{m}}|} \\ \mathbf{e}_{\mathrm{t}} &= \frac{\mathbf{x}_4^{\mathrm{m}} - \mathbf{x}_1^{\mathrm{m}}}{|\mathbf{x}_4^{\mathrm{m}} - \mathbf{x}_1^{\mathrm{m}}|} \\ \mathbf{e}_{\mathrm{n}} &= \frac{\mathbf{e}_{\mathrm{t}} \times \mathbf{e}_{\mathrm{s}}}{|\mathbf{e}_{\mathrm{t}} \times \mathbf{e}_{\mathrm{s}}|} \end{aligned}$$

Once these are determined, the transformation matrix $\mathbf{Q}$ between global and mid-plane planar system is given by:

$$\mathbf{Q} = [\mathbf{e}_{\mathrm{n}}\, \mathbf{e}_{\mathrm{t}}\, \mathbf{e}_{\ell}] \tag{A.2.0.3}$$

The main calculations are then carried out in the planar coordinate system. The mid-plane coordinates in the planar reference system $\bar{\mathbf{x}}_{\mathrm{mid}}$ are calculated as follows:

$$\bar{\mathbf{x}}_{\mathrm{mid}} = \mathbf{Q}\mathbf{x}_{\mathrm{mid}} \tag{A.2.0.4}$$

The Jacobian matrix is obtained using only the planar tangential mid-plane coordinates, hence:

$$\mathbf{J} = \begin{bmatrix} \sum_{\mathrm{i}} \overline{\mathrm{x}}^{\mathrm{m}}_{\mathrm{it}} \frac{\partial \mathrm{N_i}}{\partial \xi} & \sum_{\mathrm{i}} \overline{\mathrm{x}}^{\mathrm{m}}_{\mathrm{it}} \frac{\partial \mathrm{N_i}}{\partial \eta} \\ \sum_{\mathrm{i}} \overline{\mathrm{x}}^{\mathrm{m}}_{\mathrm{i}\ell} \frac{\partial \mathrm{N_i}}{\partial \xi} & \sum_{\mathrm{i}} \overline{\mathrm{x}}^{\mathrm{m}}_{\mathrm{i}\ell} \frac{\partial \mathrm{N_i}}{\partial \eta} \end{bmatrix} \tag{A.2.0.5}$$

The displacements of the bottom surface and of the bottom surface in global coordinates are given by:

$$\begin{aligned} \mathbf{u}^{\mathrm{bot}} &= \mathrm{N_1}\mathbf{u}_1 + \mathrm{N_2}\mathbf{u}_2 + \mathrm{N_3}\mathbf{u}_3 + \mathrm{N_4}\mathbf{u}_4 \\ \mathbf{u}^{\mathrm{top}} &= \mathrm{N_1}\mathbf{u}_5 + \mathrm{N_2}\mathbf{u}_6 + \mathrm{N_3}\mathbf{u}_7 + \mathrm{N_4}\mathbf{u}_8 \end{aligned} \tag{A.2.0.6}$$

Where the shape functions for the surface element, expressed in parent element coordinates, are given as follows:

$$\begin{aligned} &\mathrm{N_1}(\xi,\eta) = \frac{1}{4}(1-\xi)(1-\eta); \qquad \mathrm{N_2}(\xi,\eta) = \frac{1}{4}(1+\xi)(1-\eta); \\ &\mathrm{N_3}(\xi,\eta) = \frac{1}{4}(1+\xi)(1+\eta); \qquad \mathrm{N_4}(\xi,\eta) = \frac{1}{4}(1-\xi)(1+\eta); \end{aligned} \tag{A.2.0.7}$$

As a result, the displacement jump between top and bottom surface is calculated as:

$$\begin{aligned} [\![\mathbf{u}]\!] &= \mathbf{u}^{\mathrm{top}} - \mathbf{u}^{\mathrm{bot}} \\ &= \mathbf{N}_{\mathrm{CE}}\mathbf{u}_{\mathrm{CE}} \end{aligned} \tag{A.2.0.8}$$

With $\mathbf{N}_{\mathrm{CE}}$ defined as such:

$$\mathbf{N}_{\mathrm{CE}} = \begin{bmatrix} -\mathbf{N}_1 & -\mathbf{N}_2 & -\mathbf{N}_3 & -\mathbf{N}_4 & \mathbf{N}_1 & \mathbf{N}_2 & \mathbf{N}_3 & \mathbf{N}_4 \end{bmatrix} \tag{A.2.0.9}$$

Where $\mathbf{N}_{\mathrm{i}}$ is a 3×3 matrix with $\mathrm{N_i}$ along its diagonal, with $\mathrm{i} = 1,..,4$. The separation vector in planar coordinates is obtained through transformation of the displacement jump according to:

$$\boldsymbol{\delta} = \mathbf{Q}[\![\mathbf{u}]\!] \tag{A.2.0.10}$$

The traction $\boldsymbol{\tau}$ is then calculated from the separation through the constitutive matrix of the cohesive element $\mathbf{D}_{\mathrm{CE}}$:

$$\boldsymbol{\tau} = \mathbf{D}_{\mathrm{CE}}\boldsymbol{\delta} \tag{A.2.0.11}$$

Where $\mathbf{D}_{\mathrm{CE}}$ is given by:

$$\mathbf{D}_{\mathrm{CE}} = \begin{bmatrix} \mathrm{K_n} & 0 & 0 \\ 0 & \mathrm{K_t} & 0 \\ 0 & 0 & \mathrm{K}_\ell \end{bmatrix} \tag{A.2.0.12}$$

With $\mathrm{K_n}$, $\mathrm{K_t}$ and K_ℓ being the penalty stiffness in the normal, transverse tangential and longitudinal tangential directions, respectively.

Finally, the element's stiffness matrix and internal force vector are calculated according to the followings:

$$\begin{aligned}
\mathbf{K}_{\mathrm{CE}} &= \sum_{\mathrm{k}=1}^{\mathrm{n_Q}} \left(\mathbf{QN}_{\mathrm{CE}}(\boldsymbol{\xi}_{\mathrm{k}})\right)^{\mathrm{T}} \mathbf{D}_{\mathrm{CE}}(\mathbf{QN}_{\mathrm{CE}}(\boldsymbol{\xi}_{\mathrm{k}}))\mathrm{det}\mathbf{J}\mathrm{w}(\boldsymbol{\xi}_{\mathrm{k}}) \\
\mathbf{f}^{\mathrm{int}} &= \sum_{\mathrm{k}=1}^{\mathrm{n_Q}} \left(\mathbf{QN}_{\mathrm{CE}}(\boldsymbol{\xi}_{\mathrm{k}})\right)^{\mathrm{T}} \boldsymbol{\tau}\mathrm{det}\mathbf{J}\mathrm{w}(\boldsymbol{\xi}_{\mathrm{k}}) \qquad \text{where} \qquad \boldsymbol{\xi}_{\mathrm{k}} = (\xi_{\mathrm{k}}, \eta_{\mathrm{k}})
\end{aligned} \tag{A.2.0.13}$$

Where $\mathrm{n_Q}$ is the number of integration points and $\boldsymbol{\xi}$ are the natural coordinates of the integration points.

A.3. Flowcharts

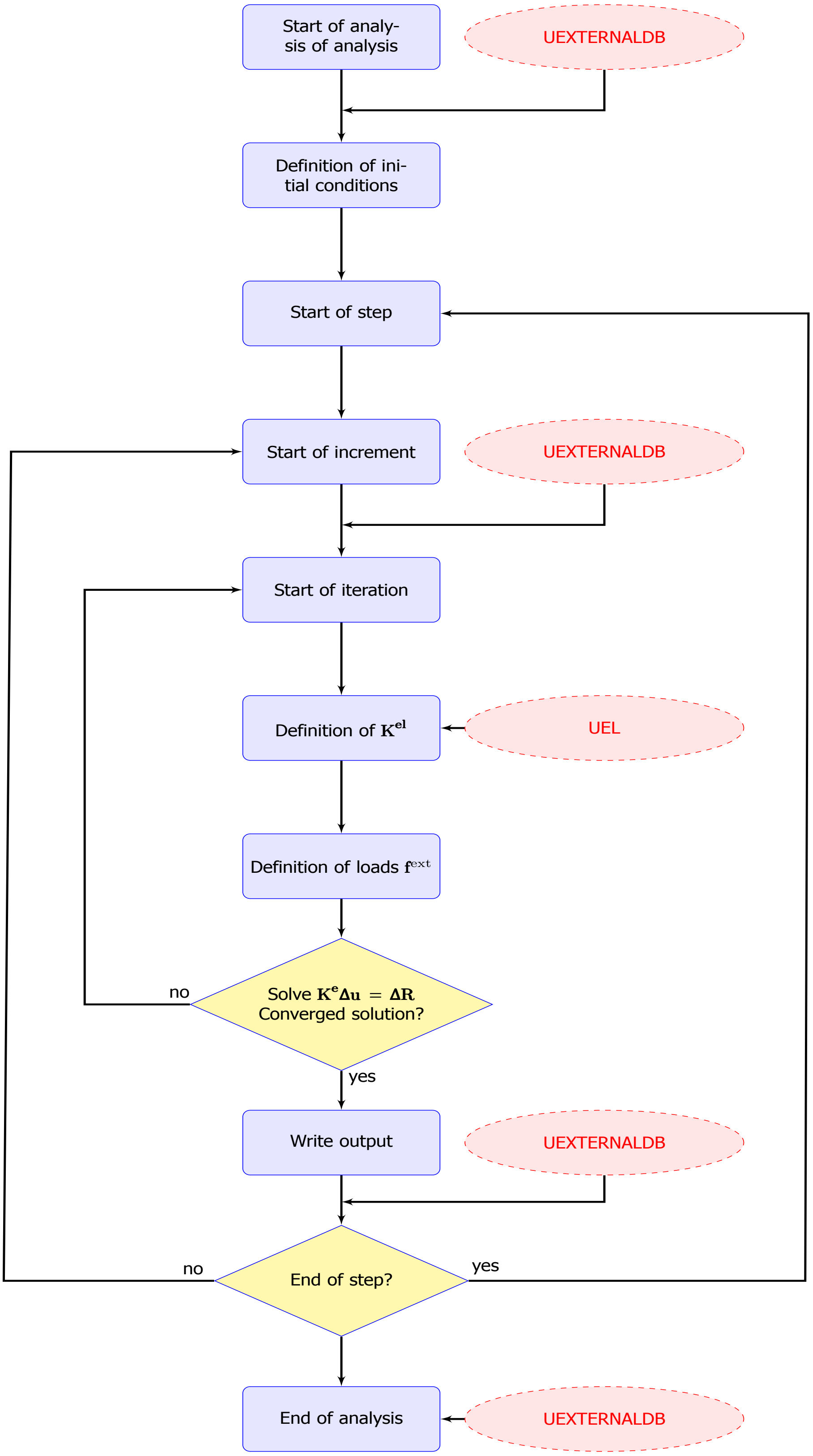


Figure A.6: Flowchart for the Abaqus/Standard procedure

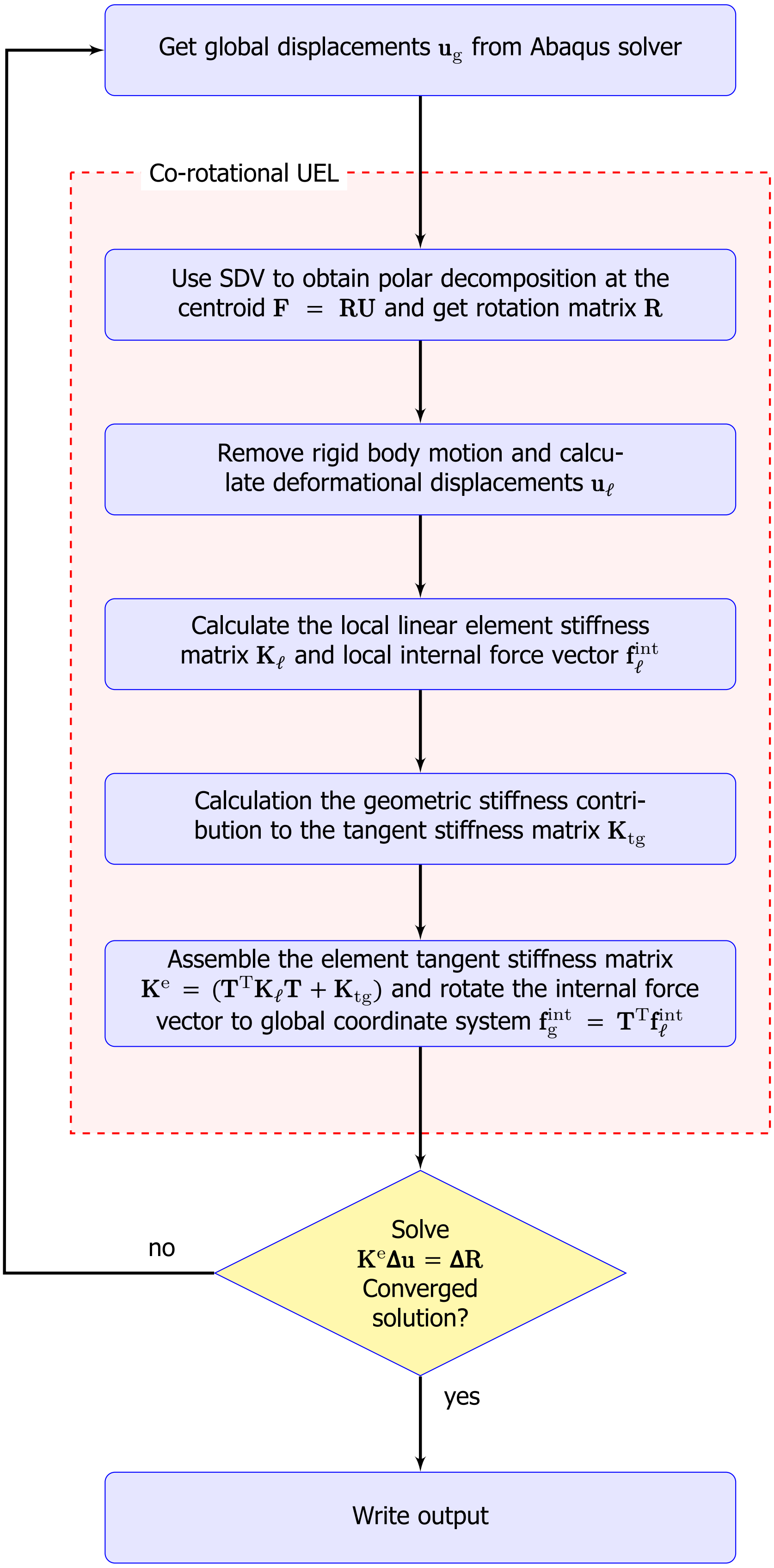


Figure A.7: UEL flowchart for the co-rotational procedure

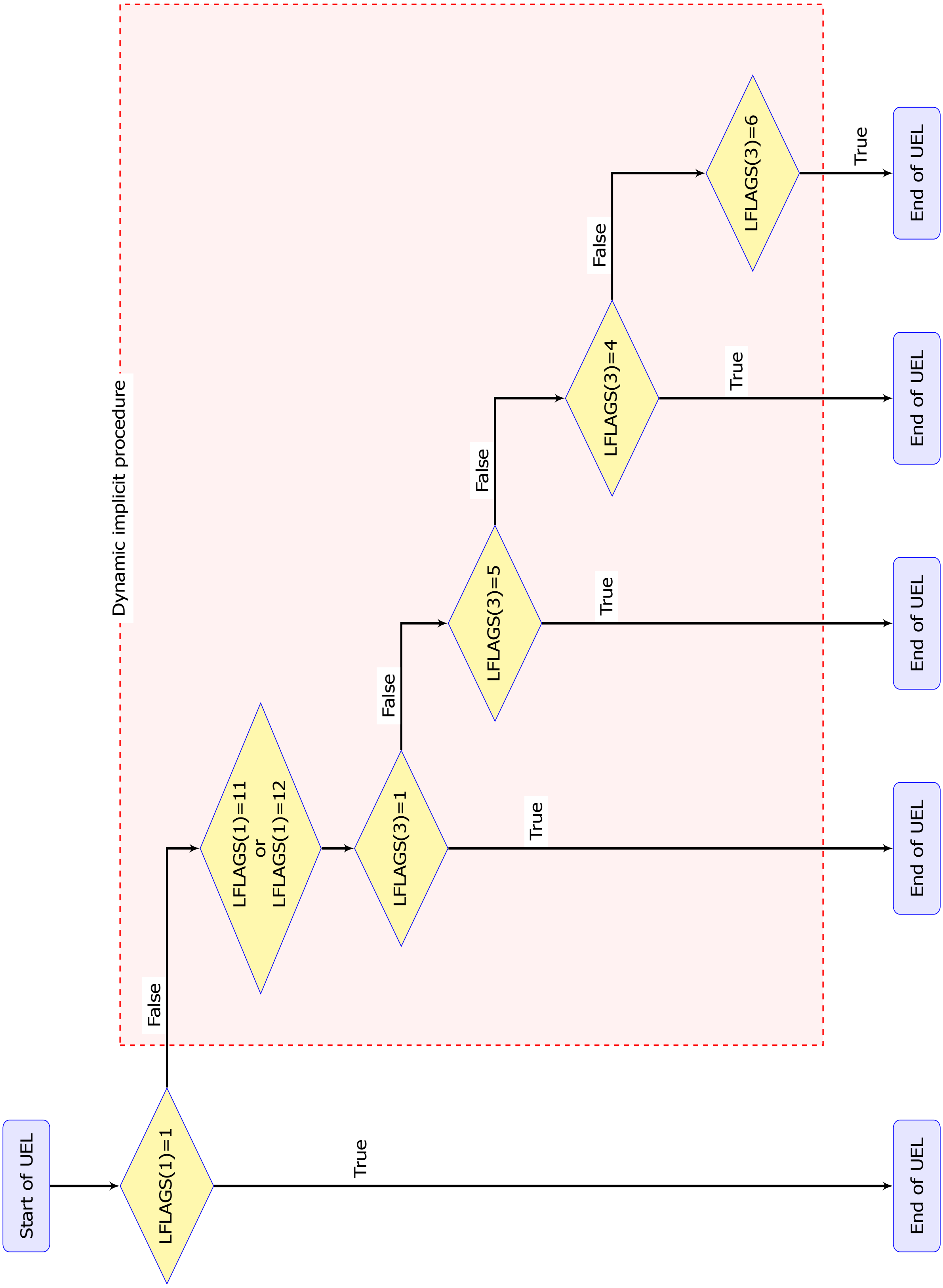


Figure A.8: UEL extension flowchart for the dynamic implicit procedure

Bibliography

[1] D. Siromani, J. Awerbuch, and T. M. Tan, *Finite element modeling of the crushing behavior of thin-walled CFRP tubes under axial compression,* Composites Part B: Engineering **64**, 50 (2014).

[2] T. Belytschko, W. K. Liu, and B. Moran, *Nonlinear Finite Elements For Continua And Structures*, 2nd ed. (John Wiley & Sons, Inc., New York, 2014).

[3] C. A. Felippa and B. Haugen, *A unified formulation of small-strain corotational finite elements: I. Theory,* Computer Methods in Applied Mechanics and Engineering **194**, 2285 (2005).

[4] I. M. Daniel, *Failure Mechanisms in Fiber-Reinforced Composites,* Proceedings of the ARPA/AFML Review of Progress in Quantitative NDE **34**, 205 (1978).

[5] B. G. Green, M. R. Wisnom, and S. R. Hallett, *An experimental investigation into the tensile strength scaling of notched composites,* Composites Part A: Applied Science and Manufacturing **38**, 867 (2007).

[6] R. D. Wood and O. C. Zienkiewicz, *Geometrically nonlinear finite element analysis of beams, frames, arches and axisymmetric shells,* Computers and Structures **7**, 725 (1977).

[7] R. D. Wood, *A total lagrangian geometrically non linear analysis of combined beam and cable structures,* Computers & Structures **17**, 115 (1983).

[8] K. J. Bathe, E. Ramm, and E. L. Wilson, *Finite element formulations for large deformation dynamic analysis,* International Journal for Numerical Methods in Engineering **9**, 353 (1975).

[9] K. J. Bathe and S. Bolourchi, *Large displacement analysis of three-dimensional beam structures,* International Journal for Numerical Methods in Engineering **14**, 961 (1979).

[10] H. D. Hibbitt, P. V. Marcal, and J. R. Rice, *A finite element formulation for problems of large strain and large displacement,* International Journal of Solids and Structures **6**, 1069 (1970).

[11] P. K. Larsen and E. P. Popov, *Large displacement analysis of viscoelastic shells of revolution,* Computer Methods in Applied Mechanics and Engineering **3**, 237 (1974).

[12] P. V. Marcal, *The effect of initial displacements on problems of large deflection and stability,* Tech report ARPA E54 (1967).

[13] P. Sharifi and D. N. Yates, *Nonlinear Thermo-Elastic-Plastic and Creep Analysis by the Finite-Element Method,* AIAA Journal **12**, 1210 (1974).

[14] L. H. Teh and M. J. Clarke, *Co-rotational and Lagrangian formulations for elastic three-dimensional beam finite elements,* Journal of Constructional Steel Research **48**, 123 (1998).

[15] J. H. Argyris, *An excursion into large rotations,* Computer Methods in Applied Mechanics and Engineering **32**, 85 (1982).

[16] M. A. Crisfield, *A consistent co-rotational formulation for non-linear, three-dimensional, beam-elements,* Computer Methods in Applied Mechanics and Engineering **81**, 131 (1990).

[17] L. Jiang, M. W. Chernuka, and N. G. Pegg, *A co-rotational, updated Lagrangian formulation for geometrically nonlinear finite element analysis of shell structures,* Finite Elements in Analysis and Design **18**, 129 (1994).

[18] G. F. Moita and M. A. Crisfield, *A finite element formulation for 3-D continua using the co-rotational technique,* International Journal for Numerical Methods in Engineering **39**, 3775 (1996).

[19] G. A. Wempner, *Finite elements, finite rotations and small strains of flexible shells,* International Journal of Solids and Structures **5**, 117 (1969).

[20] T. Belytschko and L. W. Glaum, *Applications of higher order corotational stretch theories to nonlinear finite element analysis,* Computers and Structures **10**, 175 (1979).

[21] M. A. Crisfield and G. F. Moita, *A co-rotational formulation for 2-D continua including incompatible modes,* International Journal for Numerical Methods in Engineering **39**, 2619 (1996).

[22] R. De Borst, M. A. Crisfield, J. C. Remmers, and C. V. Verhoosel, *Nonlinear Finite Element Analysis of Solids and Structures* (2012) p. 544, arXiv:arXiv:1011.1669v3 .

[23] J. H. Argyris, O. Hilpert, G. A. Malejannakis, and D. W. Scharpf, *On the geometrical stiffness of a beam in space-a consistent V.W. approach,* Computer Methods in Applied Mechanics and Engineering **20**, 105 (1979).

[24] C. C. Rankin and F. A. Brogan, *An element independent corotational procedure for the treatment of large rotations,* Journal of pressure vessel technology **108**, 165 (1986).

[25] K. M. Hsiao, J. Y. Lin, and W. Y. Lin, *A consistent co-rotational finite element formulation for geometrically nonlinear dynamic analysis of 3-D beams,* Computer Methods in Applied Mechanics and Engineering **169**, 1 (1999).

[26] B. Nour-Omid and C. C. Rankin, *Finite rotation analysis and consistent linearization using projectors,* Computer Methods in Applied Mechanics and Engineering **93**, 353 (1991).

[27] R. de Borst, J. J. C. Remmers, A. Needleman, and M. A. Abellan, *Discrete vs smeared crack models for concrete fracture: Bridging the gap,* International Journal for Numerical and Analytical Methods in Geomechanics **28**, 583 (2004).

[28] B. Y. Chen, P. M. Baiz, S. T. Pinho, and T. E. Tay, *An Extended Phantom Node Method for Crack Interactions in Composites,* in *Proceedings of the 20th UK conference of the association for computational mechanics in engineering* (Manchester, 2012).

[29] R. de Borst, *Numerical aspects of cohesive-zone models,* Engineering Fracture Mechanics **70**, 1743 (2003).

[30] K. Park and G. H. Paulino, *Cohesive Zone Models : A Critical Review of Traction-Separation Relationships Across Fracture Surfaces,* Applied Mechanics Reviews **64**, 20 (2013).

[31] Y. Mi, M. A. Crisfield, and G. Davies, *Progressive delamination using interface elements,* Journal of Composite Materials **32**, 1246 (1997).

[32] T. Belytschko and T. Black, *Elastic Crack Growth in Finite Elements With Minimal Remeshing,* Int. J. Numer. Meth. Engng **45**, 601 (1999).

[33] N. Moës, J. Dolbow, and T. Belytschko, *A finite element method for crack growth without remeshing,* International Journal for Numerical Methods in Engineering **46**, 131 (1999), arXiv:(SICI)1097-0207(19990910)46:1<131::AID-NME726>3.0.CO;2-J [0.1002] .

[34] B. Y. Chen, S. T. Pinho, N. V. De Carvalho, P. M. Baiz, and T. E. Tay, *A floating node method for the modelling of discontinuities in composites,* Engineering Fracture Mechanics **127**, 104 (2014).

[35] T. P. Fries and T. Belytschko, *The extended/generalized finite element method: An overview of the method and its application,* International Journal for Numerical Methods in Engineering **84**, 253 (2010), arXiv:1010.1724 .

[36] E. V. Iarve, M. R. Gurvich, D. H. Mollenhauer, C. A. Rose, and C. G. Davilla, *Mesh-independent matrix cracking and delamination modeling in laminated composites,* International Journal for Numerical Methods in Engineering **88**, 749 (2010), arXiv:1010.1724 .

[37] L. F. Kawashita, A. Bedos, and S. R. Hallett, *Modelling mesh independent transverse cracks in laminated composites with a simplified cohesive segment method,* Computers, Materials and Continua **32**, 133 (2012).

[38] N. Sukumar, Z. Y. Huang, J. H. Prévost, and Z. Suo, *Partition of unity enrichment for bimaterial interface cracks,* International Journal for Numerical Methods in Engineering **59**, 1075 (2004).

[39] D. B. P. Huynh and T. Belytschko, *The extended finite element method for fracture in composite materials,* International Journal for Numerical Methods in Engineering **77**, 214 (2009), arXiv:1010.1724 .

[40] H. W. Wang, H. W. Zhou, H. W. Ji, and X. C. Zhang, *Application of extended finite element method in damage progress simulation of fiber reinforced composites,* Materials and Design **55**, 191 (2014).

[41] T. Hettich, A. Hund, and E. Ramm, *Modeling of failure in composites by X-FEM and level sets within a multiscale framework,* Computer Methods in Applied Mechanics and Engineering **197**, 414 (2008).

[42] A. Hansbo and P. Hansbo, *A finite element method for the simulation of strong and weak discontinuities in solid mechanics,* Computer Methods in Applied Mechanics and Engineering **193**, 3523 (2004).

[43] P. M. A. Areias and T. Belytschko, *A comment on the article "A finite element method for simulation of strong and weak discontinuities in solid mechanics" by A. Hansbo and P. Hansbo [Comput. Methods Appl. Mech. Engrg. 193 (2004) 3523-3540],* Computer Methods in Applied Mechanics and Engineering **195**, 1275 (2006).

[44] R. de Borst, J. J. C. Remmers, and A. Needleman, *Mesh-independent discrete numerical representations of cohesive-zone models,* Engineering Fracture Mechanics **73**, 160 (2006).

[45] J. J. C. Remmers, *Discontinuities in Materials and Structures: A Unifying Computational Approach,* Ph.D. thesis, Delft University (2006).

[46] X. J. Fang, Q. D. Yang, B. N. Cox, and Z. Q. Zhou, *An augmented cohesive zone element for arbitrary crack coalescence and bifurcation in heterogeneous materials,* International Journal for Numerical Methods in Engineering **88**, 841 (2011), arXiv:1010.1724 .

[47] D. Ling, Q. D. Yang, and B. Cox, *An augmented finite element method for modeling arbitrary discontinuities in composite materials,* International Journal of Fracture **156**, 53 (2009).

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

[49] F. P. van der Meer, L. J. Sluys, S. R. Hallett, and M. R. Wisnom, *Computational modeling of complex failure mechanisms in laminates,* Journal of Composite Materials **46**, 603 (2011).

[50] N. V. De Carvalho, B. Y. Chen, S. T. Pinho, J. G. Ratcliffe, P. M. Baiz, and T. E. Tay, *Modeling delamination migration in cross-ply tape laminates,* Composites Part A: Applied Science and Manufacturing **71**, 192 (2015).

[51] B. Y. Chen, T. E. Tay, S. T. Pinho, and V. B. C. Tan, *Modelling the tensile failure of composites with the floating node method,* Computer Methods in Applied Mechanics and Engineering **308**, 414 (2016).

[52] O. C. Zienkiewicz, R. L. Taylor, K. Morgan, O. Hassan, and N. P. Weatherill, *Academy of Engineering Polish Academy of Science Chinese Academy of Sciences National Academy of Science Italy (Academia dei Lincei)*, Vol. 35 (1981) pp. 110–114.

[53] W. M. Lai, D. Rubin, and E. Krempl, *Introduction to Continuum Mechanics,* Elsevier (2010), 10.1016/B978-0-444-42021-3.50002-1, arXiv:arXiv:1011.1669v3 .

[54] K. J. Bathe, *Finite Element Procedures* (Prentice Hall, 1996).

[55] E. Gal and R. Levy, *Geometrically nonlinear analysis of shell structures using a flat triangular shell finite element,* Archives of Computational Methods in Engineering **13**, 331 (2006).

[56] K. Sze, X. Liu, and S. Lo, *Popular benchmarks for geometrically non linear problems,* Finite Elements in Analysis and Design **40**, 1551 (2004), arXiv:/linkinghub.elsevier.com/retrieve/pii/S0960982205000989 [http:] .

[57] Dassault Systèmes Simulia, *Abaqus 6.12*, Tech. Rep. (2012).

[58] K. E. Bisshopp and D. C. Drucker, *Large deflection of cantilever beams,* Quarterly of Applied Math **3**, 272 (1945), arXiv:0805.3114 .

[59] Dassault Systèmes Simulia, *Analysis User's Manual,* Abaqus 6.12 (2012).

[60] Dassault Systèmes Simulia, *Abaqus User Subroutines Reference Manual*, Tech. Rep. (2014).

[61] Kitware, *VTK File Formats,* (2012).

[62] O. C. Zienkiewicz, R. L. Taylor, and D. Fox, *The Finite Element Method for Solid and Structural Mechanics*, Vol. 2 (2014) pp. 393–447.

[63] R. Spurrier, *Comment on "Singularity-Free Extraction of a Quaternion from a Direction-Cosine Matrix",* Journal of Spacecraft and Rockets **15**, 255 (1978).

[64] P. Jetteur and S. Cescotto, *A mixed finite element for the analysis of large inelastic strains,* International Journal for Numerical Methods in Engineering **31**, 229 (1991).

[65] G. H. Golub and C. Reinsch, *Singular value decomposition and least squares solutions,* Numerische Mathematik **14**, 403 (1970).

[66] N. V. De Carvalho, B. Y. Chen, S. T. Pinho, P. M. Baiz, J. Ratcliffe, and T. E. Tay, *Floating Node Method and Virtual Crack Closure Technique for Modeling Matrix Cracking-Delamination Interaction*, Tech. Rep. July (NASA/CR–2013-218022, 2013).

[67] B. Y. Chen, T. E. Tay, S. T. Pinho, and V. B. Tan, *Modelling delamination migration in angle-ply laminates,* Composites Science and Technology **142**, 145 (2017).

[68] M. J. Laffan, S. T. Pinho, P. Robinson, and L. Iannucci, *Measurement of the in situ ply fracture toughness associated with mode I fibre tensile failure in FRP. Part I: Data reduction,* Composites Science and Technology **70**, 606 (2010).

[69] A. Rabiee and H. Ghasemnejad, *Progressive Crushing of Polymer Matrix Composite Tubular Structures: Review,* Open Journal of Composite Materials **07**, 14 (2017).

[70] D. Hull, *A Unified Approach to Progressive Crushing of Fiber-Reinforced Composite Tubes,* Composites Science and Technology **40**, 377 (1991).

[71] S. Cauchi Savona and P. J. Hogg, *Effect of fracture toughness properties on the crushing of flat composite plates,* Composites Science and Technology **66**, 2317 (2006).

[72] A. Turon, C. Dávila, P. Camanho, and J. Costa, *An Engineering Solution for Solving Mesh Size Effects in the Simulation of Delamination with Cohesive Zone Models,* Nasa Langley Research Center **TM** (2005).

[73] C. H. Huang and Y. J. Lee, *Experiments and simulation of the static contact crush of composite laminated plates,* Composite Structures **61**, 265 (2003).

[74] J. Schön, *Coefficient of friction for aluminum in contact with a carbon fiber epoxy composite,* Tribology International **37**, 395 (2004).

[75] J. Sun, K. Lee, and H. Lee, *Comparison of implicit and explicit finite element methods for dynamic problems,* Journal of Materials Processing Technology **105**, 110 (2000).

[76] T. Y. Lee, K. J. Chung, and H. Chang, *A new implicit dynamic finite element analysis procedure with damping included,* Engineering Structures **147**, 530 (2017).

[77] J. Chung and G. M. Hulbert, *a Time Integration Algorithm for Structural Dynamics With Improved Numerical Dissipation: the Generalized-A Method,* Journal of Applied Mechanics **60**, 371 (1993).

[78] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The Finite Element Method : Its Basis and Fundamentals,* sixth edit ed. (Elsevier, 2005) p. 802.