Isogeometric cohesive elements for two and three dimensional composite delamination analysis

Vinh Phu Nguyen\textsuperscript{a,1}, Pierre Kerfriden\textsuperscript{a,2}, Stéphane P.A. Bordas\textsuperscript{a,3,*}

\textsuperscript{a}School of Engineering, Institute of Mechanics and Advanced Materials, Cardiff University, Queen’s Buildings, The Parade, Cardiff CF24 3AA

Abstract

Isogeometric cohesive elements are presented for modeling two and three dimensional delaminated composite structures. We exploit the knot insertion algorithm offered by NURBS (Non Uniform Rational B-splines) to generate cohesive elements along delamination planes in an automatic fashion. A complete computational framework is presented including pre-processing, processing and post-processing. They are explained in details and implemented in MIGFEM—an open source Matlab Isogemetric Analysis code developed by the authors. The composite laminates are modeled using both NURBS solid and rotation-free shell elements. Several two and three dimensional examples ranging from standard delamination tests (the mixed mode bending test) to the L-shaped specimen with a fillet, three dimensional (3D) double cantilever beam and a 3D singly curved thick-walled laminate are provided. To the authors’ knowledge, it is the first time that NURBS-based isogeometric analysis for two/three dimensional delamination modeling is presented. IGA provides a bi-directional system in which one can go forward from CAD to analysis and backwards from analysis to CAD. This is believed to facilitate the design of composite structures.

Keywords: isogeometric analysis (IGA), B-spline, NURBS, finite elements (FEM), CAD, delamination, composite, cohesive elements, interface elements

1. Introduction

Laminated composite materials are often used in fields like automotive, aerospace, and sport equipments due to their high strength and stiffness in combination with low density. The caveat with laminated composite structures is their low out-of-plane strength and delamination or interfacial cracking between composite layers is unarguably one of the predominant modes of failure in laminated composite which can change the structural stiffness significantly and is difficult to detect during inspection. This failure mode has therefore been widely investigated both experimentally and numerically. Delamination analyses have been traditionally performed using standard low order Lagrange elements in a FEM framework, see e.g., [1, 2, 3, 4] and references therein. The two most popular computational methods for the analysis of delamination are the Virtual Crack Closure Technique (VCCT) \cite{5, 6} and interface elements with a cohesive law (also known as decohesion elements) \cite{1, 2, 3, 7}. The latter is adopted in this contribution for it can deal with initiation and propagation of delamination in a unified theory. It should be emphasized that inserting interface elements

*Corresponding author

\textsuperscript{1}nguyenpv@cardiff.ac.uk, ORCID: 0000-0003-1212-8311
\textsuperscript{2}pierre@cardiff.ac.uk
\textsuperscript{3}stephane.bordas@alum.northwestern.edu, ORCID: 0000-0001-7622-2193

Preprint submitted to Elsevier May 17, 2013
into a Lagrange finite element (FE) mesh is a time-consuming task even with commercial FE packages. Due to that fact, an open source mesh generator for cohesive modeling was developed by the first author and presented in [8]. Beside FEM, other numerical discretization methods have been utilized for delamination analyses as well. The Element Free Galerkin, which is a meshfree method, with the smooth moving least square basis was also adopted for delamination analysis [9]. In order to alleviate the computational expense of cohesive elements, formulations with enrichment of the FE basis was proposed in [10, 11]. The extended finite element method (XFEM) [12] has been adopted for delamination studies e.g., [13, 14, 15, 16] which makes the pre-processing simple for the delaminations can be arbitrarily located with respect to the FE mesh. The interaction between the delamination plane and the mesh is resolved during the solving step by using enrichment functions. However, implementation of XFEM is more involved than other numerical methods. For all aforementioned methods, curved geometries of the solids are not exactly represented.

Isogeometric analysis (IGA) was proposed by Hughes and his co-workers [17] in 2005 to reduce the gap between Computer Aided Design (CAD) and Finite Element Analysis (FEA). The idea is to use CAD technology such B-splines, NURBS (Non Uniform Rational B-splines), T-splines etc. as basis functions in a finite element (FE) framework. Since this seminal paper, a monograph was published entirely on the subject [18] and applications were found in several fields including structural mechanics, solid mechanics, fluid mechanics and contact mechanics. It should be emphasized that the idea of using CAD technologies in finite elements dates back at least to [19] where B-splines were used as shape functions in FEM and subdivision surfaces were adopted to model shells [20].

Due to the ultra smoothness provided by NURBS basis, IGA was successfully applied to many engineering problems ranging from contact mechanics, see e.g., [21, 22, 23], optimisation problems [24, 25], structural mechanics [26, 27, 28], structural vibration [29, 30], to fluids mechanics [31, 32], fluid-structure interaction problems [33]. In addition, due to the ease of constructing high order continuous basis functions, IGA has been used with great success in solving PDEs that incorporate fourth order (or higher) derivatives of the field variable such as the Hill-Cahnard equation [34], explicit gradient damage models [35] and gradient elasticity [36]. IGA has been implemented in commercial FE packages– Abaqus [37] and LS-Dyna [38]. In the context of fracture mechanics, IGA has been applied to fracture using the partition of unity method (PUM) [39] to capture two dimensional strong discontinuities and crack tip singularities efficiently [40, 41]. A phase field model for dynamic fracture has been presented in [42] where adaptive refinement with T-splines provides an effective method for simulating fracture in three dimensions. Cohesive fracture modeling in an IGA framework was presented in [43]. The method hinges on the ability to specify the continuity of NURBS/T-splines through a process known as knot insertion. Highly accurate stress fields in cracked specimens were obtained with coarse meshes. We refer to [44] for an overview of IGA and its implementation aspects.

More recently, in [45] high order B-splines cohesive FEAs with $C^0$ continuity across element boundary were utilized to efficiently model delamination of two dimensional (2D) composite specimens. In the referred paper, it was shown that by using high order B-spline (order of up to 4) basis functions along the delamination path, relatively coarse meshes can be used and 2D delamination benchmark tests such as the mixed mode bending test were solved within 10 seconds on a laptop. In this manuscript, prompted by our previous encouraging results reported in [45] plus the work in [43] and a practical motivation of a predictive tool not only for analyzing but also for designing composite laminates, we present an isogeometric framework for two and three dimensional (2D/3D) delamination analysis of laminated composites. Both the geometry and the displacement field are approximated using NURBS, therefore curved geometries are exactly represented. We use knot insertion algorithm of NURBS to duplicate control points along the delamination paths where delamination will take place. Meshes of zero-thickness interface elements can be straightforwardly generated. The proposed ideas are implemented in our open source Matlab IGA code, MIGFEM\(^4\), described in [44]. Several examples are provided including the mixed mode bending test, a L-shaped curved composite specimen test [46, 47], 3D double cantilever beam and a 3D singly curved thick-walled laminate. Moreover, isogeometric shell elements are

\(^4\)available for download at https://sourceforge.net/projects/cmcodes/
used for the first time, at least to the authors’ knowledge, to model delamination. Our findings are (i) the proposed IGA-based framework reduces significantly the time being spent on the pre-processing step to prepare FE models for delamination analyses and (ii) from the analysis perspective, the smooth high order NURBS basis functions are able to produce highly accurate stress fields which is very important in fracture modeling. The consequence is that relatively coarse meshes (compared to meshes of lower order elements) can be adopted and thus the computational expense is reduced [43, 45]. Moreover, IGA provides a bi-directional system in which one can go forward from CAD to analysis and backwards from analysis to CAD. This is believed to facilitate the design of composite structures.

The remainder of the paper is organized as follows. Section 2 gives the strong and weak formulations of the studied problem. It also points out the key difficulties of standard Lagrange finite elements used for delamination analyses. Section 3 briefly presents NURBS curves, surfaces and solids. Section 4 is devoted to a discussion on knot insertion and automatic generation of cohesive interface elements followed by finite element formulations for solids with cohesive cracks given in Section 5. Numerical examples are given in Section 6. Finally, Section 7 ends the paper with some concluding remarks.

2. Problem description

2.1. Strong form

Considering a solid \( \Omega \), as shown in Fig. (1), that is bounded by \( \Gamma \) and contains a cohesive crack \( \Gamma_d \). Prescribed displacements \( \bar{u} \) are imposed on the Dirichlet boundary \( \Gamma_u \) and prescribed tractions \( \bar{t} \) are applied on the Neumann boundary \( \Gamma_t \). Under the assumption of small displacements and gradients (note that finite deformation theories can be used in the proposed framework without any difficulties), the deformation of the material is characterized by the infinitesimal strain tensor \( \epsilon_{ij} = \frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}) \) for the bulk and the displacement jump \( [u] \) for the cohesive crack. The governing equations for quasi-static problems include the balance of linear momentum, the natural, essential boundary conditions and the traction continuity on the crack surface

\[
\begin{align*}
\nabla \cdot \sigma + b &= 0 \quad x \in \Omega & (1a) \\
\mathbf{n} \cdot \sigma &= \mathbf{t} \quad x \in \Gamma_t & (1b) \\
u &= \bar{u} \quad x \in \Gamma_u & (1c) \\
^+_{d} \mathbf{n} \cdot \sigma &= t^+_c; \quad ^-_{d} \mathbf{n} \cdot \sigma &= t^-_c; \quad t^+_c = -t^-_c = t_c \quad x \in \Gamma_d & (1d)
\end{align*}
\]

where \( \sigma \) is the Cauchy stress tensor, \( \mathbf{b} \) is the body force vector, \( \mathbf{n} \) denotes the normal to the boundary \( \Gamma \) and \( t^c \) is the cohesive traction across the crack \( \Gamma_d \) with unit normal vector \( \mathbf{n}_d \).

Constitutive equations for the bulk and the cohesive crack can be written as

\[
\sigma = \sigma(\epsilon(u), \alpha), \quad t^c = t^c([u], \beta)
\]

where \( \alpha \) and \( \beta \) are history variables. Concrete constitutive models used in this paper will be presented later, see Section 5.3.

2.2. Weak form

The weak formulation reads: finding the displacement field \( u \) such that

\[
\int_{\Omega} \delta u \cdot b \, d\Omega + \int_{\Gamma_t} \delta u \cdot \mathbf{t} \, d\Gamma_t = \int_{\Omega} \delta \epsilon : \sigma(u) \, d\Omega + \int_{\Gamma_d} \delta [u] \cdot t^c([u]) \, d\Gamma_d
\]

be satisfied for any admissible displacement field \( \delta \mathbf{u} \) subject to the Dirichlet boundary conditions on \( \Gamma_u \).

3
2.3. Difficulties with standard interface elements

For problems in which the crack path is known \textit{a priori} such as delamination in composite laminates and debonding of the matrix/inclusion interface, interface elements, which are elements inserted at the common boundary of continuum elements (Fig. (2)) where potential fracture will occur, are usually the method of choice. The reason behind the popularity of cohesive interface elements is probably due to the straightforward computer implementation.

![Figure 1: A two dimension solid containing a cohesive crack.](image)

![Figure 2: Discretization of the solid into continuum elements and zero-thickness interface elements.](image)

Interface elements, when applied to delamination analyses, suffer from two shortcomings namely (i) a long and cumbersome pre-processing step (doubling nodes along each delamination path, modifying the connectivity of continuum elements above and below the delamination path) and (ii) a refined mesh has to be employed otherwise unphysical oscillations in the global load-displacement behaviour of the structure are observed (in the worse case, this can cause the iterative solver to diverge). Furthermore, imagine that during the design process of composite laminates, the analyst decides to change a geometry parameter, he/she then has to go back to the CAD system to change the geometry model and repeat the time-consuming mesh generation procedure again.

In what follows, we present an isogeometric framework for delamination analyses that resolve all the aforementioned
shortcomings of Lagrange finite elements. We will demonstrate how straightforward it is to insert interface elements in a NURBS mesh for both 2D and 3D. And since there is a two-way link between CAD and FEA using NURBS, changes to the geometry can be dealt with without difficulties. As far as the mesh density requirement is concerned, the highly smooth NURBS basis functions can alleviate this to some extent as pointed out previously in [45].

3. NURBS curves, surfaces and solids

In this section, NURBS are briefly reviewed. We refer to the standard textbook [48] for details. A knot vector is a sequence in ascending order of parameter values, written $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}$ where $\xi_i$ is the $i$th knot, $n$ is the number of basis functions and $p$ is the order of the B-spline basis. Open knots in which the first and last knots appear $p+1$ times are standard in the CAD literature and thus used in this manuscript i.e., $\Xi = \{\xi_1, \ldots, \xi_1, \xi_2, \ldots, \xi_{p+1}, \ldots, \xi_{n+p+1}\}$.

Given a knot vector $\Xi$, the B-spline basis functions are defined recursively starting with the zeroth order basis function ($p = 0$) given by

$$ N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases} $$

and for a polynomial order $p \geq 1$

$$ N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) $$

This is referred to as the Cox-de Boor recursion formula. Note that when evaluating these functions, ratios of the form $0/0$ are defined as zero.

Some salient properties of B-spline basis functions are (1) they constitute a partition of unity, (2) each basis function is nonnegative over the entire domain, (3) they are linearly independent, (4) the support of a B-spline function of order $p$ is $p+1$ knot spans i.e., $N_{i,p}$ is non-zero over $[\xi_i, \xi_{i+p+1}]$, (5) basis functions of order $p$ have $p - m_i$ continuous derivatives across knot $\xi_i$ where $m_i$ is the multiplicity of knot $\xi_i$ and (6) B-spline basis are generally only approximants (except at the ends of the parametric space interval, $[\xi_1, \xi_{n+p+1}]$) and not interpolants.

Figure 3 illustrates some quadratic B-splines functions defined on an open non-uniform knot vector. Note that the basis functions are interpolatory at the ends of the interval thanks to the use of open knot vectors and also at $\xi = 4$, the location of a repeated knot where only $C^0$-continuity is attained. Elsewhere, the functions are $C^1$-continuous. The ability to control continuity by means of knot insertion is particularly useful for modeling discontinuities such as cracks or material interfaces as will be presented in this paper. In general, in order to have a $C^1$-continuity at a knot, its multiplicity must be $p+1$.

NURBS basis functions are defined as

$$ R_{i,p}(\xi) = \frac{N_{i,p}(\xi)w_i}{W(\xi)} = \frac{N_{i,p}(\xi)w_i}{\sum_{j=1}^{n} N_{j,p}(\xi)w_j} $$

where $N_{i,p}(\xi)$ denotes the $i$th B-spline basis function of order $p$ and $w_i$ are a set of $n$ positive weights. Selecting appropriate values for the $w_i$ permits the description of many different types of curves including polynomials and circular arcs. For the special case in which $w_i = c$, $i = 1, 2, \ldots, n$ the NURBS basis reduces to the B-spline basis. Note that for simple geometries, the weights can be defined analytically see e.g., [48]. For complex geometries, they are obtained from CAD packages such as Rhino [49].
Figure 3: Quadratic ($p = 2$) B-spline basis functions for an open non-uniform knot vector $\Xi = \{0, 0, 0, 1, 2, 3, 4, 4, 5, 5\}$. Note the flexibility in the construction of basis functions with varying degrees of regularity.

NURBS curves, defined with basis function $R_{i,p}$ and $n$ control points $B_i \in \mathbb{R}^d$ ($d$ denotes the number of space dimensions), are written as

$$C(\xi) = \sum_{i=1}^{n} R_{i,p}(\xi)B_i \quad (7)$$

Figure 4a gives an example of a NURBS curve.

Figure 4: NURBS curve and surface: (a) NURBS curve defined with the basis given in Fig. (3) and (b) NURBS surface. The images of knots $\xi_i$ ($i = 1, \ldots, n$) divide the curve into segments which has the role of finite elements in an analysis context.

Given two knot vectors (one for each direction) $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}$ and $\mathcal{M} = \{\eta_1, \eta_2, \ldots, \eta_{m+q+1}\}$ and a
control net \( B_{i,j} \in \mathbb{R}^d \), a tensor-product NURBS surface is defined as

\[
S(\xi, \eta) = \sum_{i=1}^{n} \sum_{j=1}^{m} R_{i,j}^{p,q}(\xi, \eta) B_{i,j} \tag{8}
\]

where \( R_{i,j}^{p,q} \) are given by

\[
R_{i,j}^{p,q}(\xi, \eta) = \frac{N_i(\xi)M_j(\eta)w_{i,j}}{\sum_{i=1}^{n} \sum_{j=1}^{m} N_i(\xi)M_j(\eta)w_{i,j}} \tag{9}
\]

In practice, knot vectors usually start by zeros and end by ones. Therefore, the parameter space is a unit square \( \Xi \times \mathcal{H} = \{ 0, \ldots, 0, 1, \ldots, 1 \} \times \{ 0, \ldots, 0, 1, \ldots, 1 \} \). Equation (8) defines a map from the unit square to a surface defined in the physical space. Figure 4b gives an example of NURBS surface.

In the same manner, NURBS solids are defined as

\[
S(\xi, \eta, \zeta) = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{l} R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta) B_{i,k,j} \tag{10}
\]

where \( R_{i,j,k}^{p,q,r} \) are given by

\[
R_{i,j,k}^{p,q,r}(\xi, \eta, \zeta) = \frac{N_i(\xi)M_j(\eta)P_k(\zeta)w_{i,j,k}}{\sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{l} N_i(\xi)M_j(\eta)P_k(\zeta)w_{i,j,k}} \tag{11}
\]

Derivatives of the B-splines and NURBS basis functions can be found elsewhere e.g., [17, 18].

4. Automatic generation of cohesive elements

The B-spline basis can be enriched by \( h- \), \( p- \) refinement and combination thereof. In computer aided geometric design notation \( h- \), \( p- \) refinement are referred to as knot insertion and degree elevation. We refer to the standard textbook [48] for details. In this section, we demonstrate how to use knot insertion to generate interface elements in an automatic fashion.

4.1. Knot insertion

The meaning of knot insertion is adding a new knot into the existing knot vector without changing the shape of the curve. It should be emphasized that knot insertion does not change the B-spline curves or surfaces geometrically but a direct influence on the continuity of the approximation where knots are repeated. Let us consider a knot vector defined by \( \Xi = \{ \xi_1, \xi_2, \ldots, \xi_{n+p+1} \} \) with the corresponding control net denoted by \( B \). A new extended knot vector given by \( \bar{\Xi} = \{ \xi_1, \xi_2, \ldots, \xi_{n+m+p+1} = \xi_{n+p+1} \} \) is formed where \( m \) knots are added. The \( n+m \) new control points \( \bar{B}_i \) are formed from the original control points by [48]

\[
\bar{B}_i = \alpha_i B_i + (1 - \alpha_i) B_{i-1} \tag{12}
\]
where

\[ \alpha_i = \begin{cases} 
1 & 1 \leq i \leq k - p, \\
\xi - \xi_i & k - p + 1 \leq i \leq k \\
\xi_{i+p} - \xi_i & k + 1 \leq i \leq n + p + 2 \\
0 & \text{otherwise}
\end{cases} \]  \tag{13}

Considering a quadratic B-spline curve with knot vector
\[ \Xi = \{0, 0, 0, 0.5, 1, 1, 1\} \] and control points as shown in Fig. 5 (left). On the right of the same figure, two new knots \( \xi = 0.25 \) and \( \xi = 0.75 \) were added. Consequently, two new control points were formed. Although the curve is not changed geometrically and parametrically, the basis functions are now richer and may be more suitable for the purpose of analysis.

Let us now consider a quadratic B-spline defined using \( \Xi = [0, 0, 0, 1, 1, 1] \). The three basis functions for this curve are given in Fig. (6a). Now suppose that we need to have a discontinuity at \( \xi = 0.5 \). This can be achieved by inserting a new knot \( \xi = 0.5 \) three (= \( p + 1 \)) times. The new knot vector is then given by \( \Xi' = [0, 0, 0, 0.5, 0.5, 1, 1, 1] \) and the new basis functions are shown in Fig. (6b). Let us build a B-spline curve with the control net defined by \( \mathbf{B} \) as shown in Eq. (14). The new control net that is defined by \( \mathbf{B}' \) is also given in Eq. (14).

\[ \mathbf{B} = \begin{bmatrix} 0.0 & 0.0 \\
0.5 & 0.5 \\
1.0 & 0.0 \end{bmatrix}, \quad \mathbf{B}' = \begin{bmatrix} 0.00 & 0.00 \\
0.25 & 0.25 \\
0.50 & 0.25 \\
0.75 & 0.25 \\
1.00 & 0.00 \end{bmatrix} \]  \tag{14}

where it should be noted that \( \mathbf{B}'_3 = \mathbf{B}'_4 \). The B-spline curve corresponds to the original and new basis is the same and given in Fig. (6c). Imagine now that point \( \mathbf{B}'_4 \) slightly moves vertically, the resulting B-spline curve with a strong discontinuity at \( x = 0.5 \) is plotted in Fig. (6d). This technique of inserting knot values \( p + 1 \) times was used in [43] to model the decohesion of material interfaces. The application of this method in two/three dimensions resemble the usage of zero-thickness interface elements by doubling nodes in the standard Lagrange FE framework.

We demonstrate the technique to generate a discontinuity into a NURBS surface by a simple example. The studied surface is a square of \( 10 \times 10 \) and suppose that one needs a horizontal discontinuity line in the middle of the square
Figure 6: \( p + 1 \) times knot insertion for a quadratic B-spline curve to introduce a \( C^{-1} \) discontinuity at \( \xi = 0.5 \).

as shown in Fig. (7a). The coarsest mesh consists of one single bi-linear NURBS element with \( \Xi = \mathcal{K} = \{0, 0, 1, 1\} \) and \( p = q = 1 \). To insert the desired discontinuity, the following steps are performed: (1) perform order elevation to \( p = q = 2 \); (2) perform knot insertion for \( \mathcal{K} \), the new knot is \( \mathcal{K} = \{0, 0, 0.5, 0.5, 0.5, 1, 1, 1\} \) (Fig. (7b)); and (3) perform knot insertion to refine the mesh if needed. In Fig. (7c,d) the duplicated control points were moved upward to show the effect of discontinuity. In order to use these duplicated nodes in a FE context, one can put nonlinear springs connecting each pair of nodes, see e.g., [50] or employ zero-thickness interface elements. In this manuscript the latter is used. With a small amount of effort, the connectivity matrix for the interface elements can be constructed using a simple Matlab code as given in Listing 1. It is obvious that, due to the simplification implied by line 2 of Listing 1, this code snippet applies only for a horizontal/vertical discontinuity line. However, it is straightforward to extend this template code to general cases by changing line 2. Such refinements are certainly problem dependent and hence not provided here. We refer to Fig. (8) for one example of a curved composite panel made of two plies. Line 1 of Listing 1 builds the element connectivity array for a 1D NURBS mesh, we refer to [44] for a detailed description of these Matlab functions. Here, an assumption was made that interface elements are parallel to the \( \Xi \) knot vector.

Listing 1: Matlab code to build the element connectivity for 1D interface elements.

\begin{verbatim}
1  [ielements] = buildIGAIMesh (uKnot,p);
2  delaminationNodes = find(abs(controlPts(:,2) - 5) < 1e-10);
3  mm = 0.5*length(delaminationNodes);
\end{verbatim}
Figure 7: Example of introducing a horizontal discontinuity in a NURBS surface.

Figure 8: L-shaped composite sample of two plies with a fillet modeled with a bi-quadratic NURBS: red circles denote duplicated nodes. For this case, it suffices to find the index of node S—the first node on the discontinuity curve. By virtue of the tensor-product nature of NURBS, the indices of other discontinuity nodes can then be found with ease.

Listing 2: Matlab code to build the element connectivity for 2D interface elements.
upperNodes = delaminationNodes(mm+1:end);
% noElemsU = number of elements along X-dir
iElements = zeros(noElemsU * noElemsV, 2*(p+1)*(q+1));
iElements = generateIGA2DMesh(uKnot, vKnot, noPtsX, noPtsY, p, q);
for e = 1:noElemsU * noElemsV
    iElements(e, 1:(p+1)*(q+1)) = lowerNodes(iElements(e, :));
    iElements(e, (p+1)*(q+1)+1:end) = upperNodes(iElements(e, :));
end

The technique introduced so far can be straightforwardly extended to three dimensions, see Listing 2 and Fig. (9) for an example. The discontinuity surface lies in the $X-Y$ plane. Line 7 of this Listing builds the element connectivity array for a 2D NURBS mesh, we refer to [44] for a detailed description of these Matlab functions. These pre-processing techniques are implemented in our open source Matlab IGA code named MIGFEM, described in [44], which is available at https://sourceforge.net/projects/cmcodes/. In order to support IGA codes which are based on the Bézier extraction [51, 52], see also Section 5.5, MIGFEM computes the 1D, 2D and 3D Bézier extractors. In summary the pre-processing code writes to a file with (1) coordinates of control points (including duplicated ones), (2) connectivity of continuum elements, (3) connectivity of interface elements, (4) 2D/3D Bézier extractors for continuum elements and (5) 1D/2D extractors for interface elements.

![Discontinuity Surface](image)

Figure 9: A 3D bar with a discontinuity surface in the middle: modeled by a tri-quadratic NURBS solid.

Remark 4.1. In the proposed framework, interface elements are inserted a priori, therefore delaminations only grow along predefined paths. For laminates built up by plies of unidirectional fiber reinforced composites, the fracture toughness of the plies is much greater than the fracture toughness of the ply interfaces. Therefore, delaminations only grow along the ply interfaces which are known a priori. And that justifies our assumption. It should be emphasized that knot insertion and order elevation are very basic algorithms of NURBS and implementation can be found for example in [48]. In this work, we use the NURBS toolbox described in [53]. For complex geometries, one can use Rhino [49] and implement our algorithms as plugins for Rhino. Another option is to export Rhino NURBS geometries to files and use them on our Matlab code.
5. Finite element formulation

This section presents a finite element discretization of the weak form given in Eq. (3) using NURBS. We begin with a brief review of an isogeometric Galerkin finite element formulation of which details can be found elsewhere [17, 18, 44]. Next, FE discrete equations are given followed by a discussion on cohesive laws, numerical integration and implementation aspects.

5.1. Isogeometric analysis

According to the IGA the field variable (which is, in this paper, the displacement field) is approximated by the same B-spline/NURBS basis functions used to exactly represent the geometry. Therefore, in an IGA context, one writes for the geometry and the displacement field, respectively

\[ x = N_I(\xi)x_I \]  
\[ u_i = N_I(\xi)u_{iI} \]  

where \( x_I \) are the nodal coordinates, \( u_{iI} \) is the \( i \) \((i = 1, 2, 3)\) component of the displacement at node/control point \( I \) and \( N_I \) denotes the shape functions which are the B-spline/NURBS basis functions described in Section 3. It should be emphasized that Eq. (15a) is a global mapping—it maps from the parameter space (unit square/cube) to many elements. It is in contrary to FE mapping which is local.

Elements are defined as non-zero knot spans, see Fig. (10), which are elements in the parameter space (denoted by \( \hat{\Omega}_e \)). Their images in the physical space obtained via the mapping, see Eq. (15a), are called elements in the physical space (denoted by \( \Omega_e \)) that resemble the familiar Lagrange elements. From our experiences, it is beneficial to work with elements in the parameter space. Numerical integration is also performed on a parent domain as in Lagrange FEs. The element connectivity of NURBS elements are different from Lagrange elements, we therefore give an example in Fig. (11). For this example, the element connectivity matrix \( element \) reads

\[
\begin{bmatrix}
1 & 2 & 3 & 5 & 6 & 7 & 9 & 10 & 11 \\
2 & 3 & 4 & 6 & 7 & 8 & 10 & 11 & 12 \\
5 & 6 & 7 & 9 & 10 & 11 & 13 & 14 & 15 \\
6 & 7 & 8 & 10 & 11 & 12 & 14 & 15 & 16
\end{bmatrix}
\]  

5.2. FE discrete equations

The discrete equations of the weak form given in Eq. (3) are

\[ f_{\text{ext}} - f_{\text{int}} - f_{\text{coh}} = 0 \]  

where \( f_{\text{ext}} \) is the external force vector, the internal force vector is denoted as \( f_{\text{int}} \) and the cohesive force vector \( f_{\text{coh}} \). The elemental external and internal force vectors are computed from contributions of continuum elements and given by

\[
f_{\text{ext}} = \int_{\Omega_e} B^T \sigma d\Omega_e
\]

\[
f_{\text{int}} = \int_{\Omega_e} \rho N^T b d\Omega_e + \int_{\Gamma^e} N^T \bar{t} d\Gamma^e
\]
Figure 10: Definition of domains used for integration in isogeometric analysis. Elements are defined in the parametric space as non-zero knot spans, $[\xi_i, \xi_{i+1}] \times [\eta_j, \eta_{j+1}]$ and elements in the physical space are images of their parametric counterparts.
where $\Omega_e$ is the element domain, $\Gamma_e^N$ is the element boundary that overlaps with the Neumann boundary, $\mathbf{b}$ and $\mathbf{t}$ are the body forces and traction vector, respectively. The shape function matrix and the strain-displacement matrix are denoted by $\mathbf{N}$ and $\mathbf{B}$; $\mathbf{\sigma}$ is the Cauchy stress vector.

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

$$
\mathbf{f}_{ie, +}^{coh} = + \int_{\Gamma_{ie}} \mathbf{N}^T_{int} \mathbf{t} \mathbf{\varepsilon} d\Gamma
$$

$$
\mathbf{f}_{ie, -}^{coh} = - \int_{\Gamma_{ie}} \mathbf{N}^T_{int} \mathbf{t} \mathbf{\varepsilon} d\Gamma
$$

(20)
in which $\mathbf{t}^c$ denotes the cohesive traction, $\mathbf{N}_{\text{int}}$ represents the shape function matrix of interface elements and $\Gamma_{ic}$ is the interface element domain which is chosen to be the mid-surface of the interface element [54] that makes the formulation also valid for large displacements. The subscripts +/- denote the upper and lower faces of the interface element.

The displacement of the upper and lower faces of an interface element, let say the first element in Fig. (12)-left read

$$
\mathbf{u}^+ = N_1(\xi)\mathbf{u}_5 + N_2(\xi)\mathbf{u}_6 + N_3(\xi)\mathbf{u}_7 \\
\mathbf{u}^- = N_1(\xi)\mathbf{u}_1 + N_2(\xi)\mathbf{u}_2 + N_3(\xi)\mathbf{u}_3
$$

with $N_I$ ($I = 1, 2, 3, 4$) are the quadratic NURBS shape functions. Figure 12 also explains the difference between $C^{p-1}$ and $C^0$ high order elements–for the same number of elements, $C^{p-1}$ meshes have less nodes. We refer to [17] for more information on this issue. The latter was used in [45] with B-spline basis for 2D delamination analysis.

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

$$
[\mathbf{u}(\mathbf{x})] \equiv \mathbf{u}^+ - \mathbf{u}^- = \mathbf{N}_{\text{int}}(\mathbf{u}^+ - \mathbf{u}^-)
$$

where

$$
\mathbf{N}_{\text{int}} = \begin{bmatrix}
N_1 & 0 & N_2 & 0 & N_3 & 0 \\
0 & N_1 & 0 & N_2 & 0 & N_3
\end{bmatrix}, \quad \mathbf{u}^+ = [u_{x5} \quad u_{y5} \quad u_{x6} \quad u_{y6} \quad u_{x7} \quad u_{y7}]^T
$$
The displacement jump will be inserted into a cohesive law (or traction-separation law) to compute the corresponding traction $t_c$. We refer to [1, 2, 3] and references therein for other aspects of interface cohesive elements. The implementation for three dimensional problems i.e., 2D interface elements is straightforward, for example in Eq. (21), instead of using univariate NURBS basis one uses bivariate basis $N_I(\xi, \eta)$. Linearization of Eq. (17) required in an iterative solution scheme can be performed in a standard fashion, we refer to e.g., [55, 16, 8] for details.

5.3. Cohesive laws

In this work, we adopt the damage-based bilinear cohesive law developed in [56, 57]. This is a cohesive law in which the fracture toughness is a phenomenological function, rather than a material constant, of mode mixity as formulated by Benzeggagh and Kenane [58]. Herein we briefly recall the cohesive law of which implementation details can be found in [16]. Denoting $d$ as the damage variable ($0 \leq d \leq 1$), the cohesive law reads in the local coordinate system attached to the interface elements

$$t_c^l = (1 - d)K[u]_I$$

(24)

where $K$ is the dummy stiffness. The damage variable $d$ is a function of the equivalent displacement jump, the onset $[u]_{eq}^0$ and the propagation equivalent displacement jump $[u]_{eq}^f$. The onset $[u]_{eq}^0$ is a function of $K$, the mode mixity and the normal and shear strength $\tau^0_1$ and $\tau^0_3$. The propagation displacement jump $[u]_{eq}^f$ is a function of $[u]_{eq}^0$, mode I and II fracture toughness $G_{Ic}$, $G_{IIc}$, the mode mixity and $\eta$ which is a curve fitting value for fracture toughness tests performed by Benzeggagh and Kenane [58].

5.4. Numerical integration

In this manuscript, full Gaussian integration schemes are used. Precisely, for 2D solid elements of order $p \times q$, a $(p + 1) \times (q + 1)$ Gauss quadrature rule is adopted and for cohesive elements of order $p$, a $(p + 1)$ Gauss scheme is utilized. A similar rule was used for 3D solid elements and 2D cohesive elements.

5.5. Implementation aspects

There are at least two approaches to incorporating IGA into existing FE codes—with and without using the Bézier extraction. The former, which relies on the Bézier decomposition technique, was developed in [51, 52] and provides data structures (the so-called Bézier extractor sparse matrices) that facilitate the implementation of IGA in existing FE codes. Precisely, the shape functions of IGA elements are the Bernstein polynomials (defined on the standard parent element) multiplied by the extractors. We refer to [44] for a discussion on both techniques.

For curved geometries, the post-processing of IGA is more involved than Lagrange FEAs due to two reasons (1) some control points locate outside the physical domain (hence the computed displacements at control points are not physical nodal values) and (2) existing post-processing techniques cannot be applied directly to NURBS meshes. Interested reader can refer to [44] for a discussion on some post-processing techniques for IGA. For completeness we discuss briefly one technique here for 2D problems. First, a visualization mesh which consists of four-noded quadrilateral elements is constructed. The nodes of this mesh are the intersections of the $\xi$ and $\eta$ knot lines in the physical space. We then extrapolate the quantities at Gauss points to these nodes and perform nodal averaging if necessary. Figure 13 summarizes the idea.

B-spline/NURBS basis functions are generally only approximants and not interpolants. Therefore imposing Dirichlet boundary conditions (BCs) will generally require special care [44]. However enforcement of homogeneous Dirichlet BCs that are imposed on the boundary of NURBS solids is simply as standards as in Lagrange FEAs thanks to the use of open knot vectors.
6. Examples

Since we are introducing a computational framework for delamination analyses rather than a detailed study of the delamination behaviour of composite materials, intralaminar damage (matrix cracking and fiber damage) is not taken into account leading to an orthotropic elastic behaviour assumption for the plies. Note that matrix cracking can however be efficiently modeled using extended finite elements as shown in [55, 16] and can be incorporated in our framework without major difficulties. Besides, inertia effects are also skipped. In order to trace equilibrium curves we use either a displacement control (for problems without snapbacks) and the energy-based arc-length control [59, 60]. Interested reader can refer to [45, 55] for the computer implementation aspects of this arc-length solver. A full Newton-Raphson method was used to solve the discrete equilibrium equations. Unless otherwise stated, a geometrically linear formulation is adopted. We use a C++ code [61] for computations since Matlab is not suitable for this purpose. Whenever possible, verification against theoretical solutions are provided. However validation against experiment is not provided because we are not trying to validate any model here.

Four numerical examples are provided including

- Mixed mode bending test (MMB), 2D simple geometry, implementation verification test;
- L-shaped specimen, single and multiple delamination, NURBS curved geometry;
- 3D double cantilever beam, to verify the implementation;
- Singly curved thick-walled laminate, 3D curved geometry.

And in an extra example, we present NURBS parametrization for other commonly used composite structures—glare panel with a circular initial delamination, open hole laminate and doubly curved composite panel.

6.1. Mixed mode bending test (MMB)

Figure 14 shows the mixed mode bending test of which the geometry data are $L = 100$ mm, $h = 3$ mm; the beam thickness $B$ is equal to 10 mm; the initial crack length is $a_0 = 20$ mm. The plies are modeled with isotropic material...
to make a fair comparison with analytical solutions [62] which are valid for isotropic materials only. The properties for the isotropic material are $E = 150$ GPa and $\nu = 0.25$. The properties for the cohesive elements are $G_{IC} = 0.352$ N/mm, $G_{IIc} = 1.45$ N/mm and $\tau_1^0 = 80$ MPa, $\tau_3^0 = 60$ MPa. The interface stiffness is $K = 10^6$ N/mm$^3$ and $\eta = 1.56$.

In order to prevent interpenetration of the two arms, in addition to cohesive elements, frictionless contact elements are placed along the initial crack. The loads applied are $P_1 = 2P_c/L$ and $P_2 = P(2c + L)/L$, where $L$ is the beam length, $c$ is the lever arm length, and $P$ is the applied load. From these relationships, it is clear that the applied loads $P_1$ and $P_2$ are proportional i.e., $P_2/P_1 = (2c + L)/L$. We choose $c = 43.72$ mm so that the mixed-mode ratio $G_{I}/G_{II}$ is unity.

The external force vector is therefore $\mathbf{f}^{ext} = \lambda [1, -2.1436]^T$ (a unit force was assigned to $P_1$) in which the variable load scale $\lambda$ is solved together with the nodal displacements using the energy based arc-length method [59, 60, 45].

![Figure 14: Mixed Mode Bending (MMB): geometry and loading.](image)

6.1.1. Geometry and mesh

For those who are not familiar to B-splines/NURBS, we present how to build the beam geometry using B-splines. It is obvious that the beam can be exactly represented by a bilinear B-spline surface with 4 control points locating at its four corners. The Matlab code for doing this is lines 1–9 in Listing 3. Next, the B-spline is order elevated to the order that suits the analysis purpose, see line 10 of the same Listing. The delamination path locates in the midline of the beam i.e., $\eta = 0.5$ and note that $q = 2$, in order to introduce a discontinuity one simply has to insert 0.5 three ($= q + 1$) times into knot vector $H$ (knot vector which is perpendicular to the delamination plane). For point load $P_2$ one needs a control point at the location of the force which corresponding to insert 0.5 three times (equals $p = 3$) into knots $\Xi$. Line 13 does exactly that. In order to differentiate cohesive elements and contact elements (remind that contact elements are put along the initial crack to prevent interpenetration), a knot $1 - a_0/L$ is added to $\Xi p$ times (see line 14). The final step is to perform a $h$-refinement to refine the mesh and extract element connectivity data for the interface elements using the code given in Listing 1.

**Listing 3: Matlab code to build the beam using B-splines**

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>controlPts = zeros(4,2,2);</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>controlPts(1:2,1,1) = [0;0];</td>
<td>% L=length of beam</td>
</tr>
<tr>
<td>3</td>
<td>controlPts(1:2,2,1) = [L;0];</td>
<td>% W=height of beam</td>
</tr>
<tr>
<td>4</td>
<td>controlPts(1:2,1,2) = [0;W];</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>controlPts(1:2,2,2) = [L;W];</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>controlPts(4,:,:) = 1;</td>
<td>% weights = 1: B-splines</td>
</tr>
</tbody>
</table>

18
6.1.2. Analyses with varying basis orders

We use meshes with two elements along the thickness direction and the basis order along this direction is fixed to 2 (quadratic basis). The notation \(2 \times 128 \times 2 \times 3\) indicates a mesh of \(2 \times 128\) elements of orders \(2 \times 3\). The order of basis functions along the length direction, \(p\), varies from two to five. Firstly we perform a mesh convergence test for quartic-quadratic elements and the result is given in Fig. (15a). Mesh \(2 \times 64\) is simply too coarse to accurately capture the cohesive zone and mesh \(2 \times 128\) is sufficient to get a reasonable result. Next, the mesh density is fixed at \(2 \times 128\) and \(p\) is varied from 2 to 5, the result is plotted in Fig. (15b). We refer to [45] for a throughout study on the excellent performance of high order B-splines elements compared to low order Lagrange finite elements for delamination analyses.

![Figure 15: Mixed Mode Bending (MMB): (a) mesh convergence test and (b) varying basis order in the length direction on meshes of \(2 \times 128\) elements.](image)

6.2. L-shaped composite panel with a fillet

For the second example, we analyze the L-shaped composite specimen which was studied in [47, 46] using Lagrange finite elements. The geometry and loading configuration is given in Fig. (16). Contrary to the previous example, in this example NURBS surfaces are used to exactly represent the curved geometry (to be precise circular arcs). The structure is built up by 15 plies of a unidirectional fiber reinforced carbon/epoxy material. The plies are oriented in alternating \(0^\circ\) and \(90^\circ\) orientation, where the angle is measured from the \(xy\) plane. The inner ply and the outer ply are oriented in the \(0^\circ\) direction. Material constants are given in Table 1 which are taken from [47, 46]. A plane strain condition...
is assumed. For this problem, unless otherwise stated, we use bi-quadratic NURBS elements for the continuum and quadratic NURBS interface elements for the delamination.

![Diagram of L-shaped specimen](image)

**Figure 16**: L-shaped specimen: boundary and geometry data. There are 15 plies (0° and 90°). The ply orientation is measured with respect to the $x − y$ plane.

<table>
<thead>
<tr>
<th>$E_{11}$</th>
<th>$E_{22} = E_{33}$</th>
<th>$G_{12} = G_{13}$</th>
<th>$\nu_{12} = \nu_{13}$</th>
<th>$\nu_{23}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>139.3 GPa</td>
<td>9.72 GPa</td>
<td>5.59 GPa</td>
<td>0.29</td>
<td>0.40</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$G_{1c}$</th>
<th>$G_{11c}$</th>
<th>$\tau_1^0$</th>
<th>$\tau_3^0$</th>
<th>$\mu$</th>
<th>$K$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.193 N/mm</td>
<td>0.455 N/mm</td>
<td>60.0 MPa</td>
<td>80.0 MPa</td>
<td>2.0</td>
<td>$10^6$ N/mm$^3$</td>
</tr>
</tbody>
</table>

**Table 1**: L-shaped specimen: material properties.

### 6.2.1. Geometry and mesh

The L-shaped geometry can be exactly represented by a quadratic-linear NURBS surface as shown in Fig. (17) that consists of $7 \times 2$ control points. The Matlab code used to build the NURBS is given in Listing 4. It is easy to vary the number of plies (see line 4 of the same Listing). Listing 5 gives code to perform $p$-refinement (to a bi-quadratic NURBS surface) and knot insertion at ply interfaces (two times) to create $C^0$ lines so that the strain field is discontinuous across the ply interfaces. Next, knot insertion is performed again to generate discontinuity lines at the desired ply interfaces. Two cases are illustrated in the code–interface elements locate along the interface between ply 5 and 6 (line 10) and interface elements at every ply interfaces (line 12-16).

**Listing 4**: Matlab code to build NURBS geometry of the L-shaped specimen.

```matlab
1 H = 6.4;
2 R = 2.55;
3 R0 = 2.25;
```
Figure 17: L-shaped specimen: quadratic-linear NURBS geometry with control points (filled circles) and control polygon.

```matlab
no = 15;  \% number of plies
R0 = 10;  \% ply thickness
uKnot = [0 0 0 1 1 2 2 3 3 3];  \% initial mesh - quadratic x linear
vKnot = [0 0 1 1];
controlPts = zeros(4,7,2);
controlPts(1:2,1,1) = [H+R;0]; controlPts(1:2,1,2) = [H+R;-R0];
controlPts(1:2,2,1) = [(H+R)/2;0]; controlPts(1:2,2,2) = [R0;-R0];
controlPts(1:2,3,1) = [0;0]; controlPts(1:2,3,2) = [-R0;-R0];
controlPts(1:2,4,1) = [R;0]; controlPts(1:2,4,2) = [R0;R0];
controlPts(1:2,5,1) = [0;R]; controlPts(1:2,5,2) = [-R0;R];
controlPts(1:2,6,1) = [0;(H+R)/2]; controlPts(1:2,6,2) = [-R0;(H+R)/2];
controlPts(1:2,7,1) = [0;H+R]; controlPts(1:2,7,2) = [-R0;H+R];
fac = 1/sqrt(2);  \% all weights are units except two points
controlPts(4,:) = 1;
controlPts(4,4,1) = controlPts(4,4,2) = fac;
controlPts(1:2,4,1) = fac*controlPts(1:2,4,1);
controlPts(1:2,4,2) = fac*controlPts(1:2,4,2);
solid = nrbmak(controlPts, {uKnot,vKnot});  \% build NURBS object

Listing 5: Matlab code to generate discontinuity lines.

```
6.2.2. Single delamination with and without initial cracks

Delamination of the interface between ply five and six is first analyzed. Note that at other ply interfaces, there is no cohesive elements. Firstly, the case of no initial cracks is considered. One layer of elements is used for each ply. The contour plot of damage on the deformed shape is given in Fig. (18) and the response in terms of reaction-displacement curve is plotted in Fig. (19). There is a sharp snap-back that corresponds to an unstable delamination growth. After the delamination reaches a certain size, stable delamination growth is observed as shown by the increasing part of the load-displacement curve. This is in good agreement with the semi-analytical analysis in [47]. The excellent performance of the energy-based arc-length control for responses with sharp snap-backs has been demonstrated elsewhere e.g., [45, 55], we therefore do not give a discussion on this issue.

![Figure 18: L-shaped specimen: delamination configurations at the peak (left) and when the delamination reached the two ends (right).](image)

Let assume now that there is an initial crack lying on the interface between ply 5 and 6, see Fig. (20). The initial crack is a part of the NURBS curve that defines the interface of ply 5 and 6. In this case the geometry modeling procedure is more involved and follows the steps given in Listing 6. The extra step is to perform a point inversion algorithm [48] to find out the parameters $\xi_1$ and $\xi_2$ that correspond to points $x_1$ and $x_2$—the tips of the initial crack. After that $\xi_1, \xi_2$ are inserted twice (remind that the NURBS basis order along the $\xi$ direction is two).

Listing 6: L-shaped specimen with an initial crack: Matlab code to build the geometry.

```matlab
1 % code from Listing 4 to build the NURBS surface
2 % code from Listing 5 to create C‘0 and C‘{-1} lines
3 % point inversion to find parametric values xi1 and xi2 that correspond to
4 % points x_1 and x_2 defining the initial crack.
5 % insert xi1 and xi2 2 times (p=) to have C‘0 at x_1 and x_2
6 solid = nrbkntins(solid, {{xi1 xi1 xi2 xi2} []});
7 % h-refinement along \xi direction to have a refined model for FEA
```

Remark 6.1. Point inversion for NURBS curves concerns the computation of parameter $\tilde{\xi}$ that corresponds to a point $\bar{x}$ such that $N_I(\tilde{\xi})x_I = \bar{x}$ where $x_I$ denote the control points of the curve. Generally, a Newton-Raphson iterative method is used, we refer to [48] for details.
Figure 19: L-shaped specimen with one single delamination between ply 5 and 6: without initial cracks, with an small and large initial crack.

Figure 20: L-shaped specimen with one initial crack.
Two cases, one with a small initial crack and one with a large initial crack are considered. The delamination of the specimen is given in Fig. (21) and the responses are plotted in Fig. (19). For the case of a small initial crack, the response of the specimen is very similar to the case without any initial cracks, except that the peak load is smaller. For the case of a large initial crack, the delamination growth is stable. This is in good agreement with the work in [47].

6.2.3. Multiple delaminations
In order to study multiple delaminations, we place cohesive elements along all ply interfaces and one large initial crack at the interface of ply 3 and 4 (we conducted an analysis without any initial crack and found that delamination was initiated at the interface of ply 3 and 4). The analysis was performed using about 100 load increments and the computation time was 730s on a Intel Core i7 2.8GHz laptop (29 340 unknowns and 9280 elements). Figure 22 gives the response of the specimen. As can be seen, the propagation of the first delamination (from both tips of the initial crack) is stable and the growth of the second delamination (between ply 7 and 8) is unstable.

6.3. Three dimensional double cantilever beam
As the simplest 3D delamination problem as far as geometry is concerned, we consider the 3D double cantilever beam (DCB) problem as given in Fig. (23). This example serves as a verification test for (1) verifying the implementation of 3D isogeometric interface elements and (2) validating the automatic generation of 2D isogeometric interface elements.

6.3.1. Geometry and mesh
The beam geometry is represented by one single tri-linear NURBS (actually B-splines as the weights are all units), see Lines 1–5 of Listing 7. Order elevation was then performed to obtain a tri-quadratic solid (line 7) followed by a knot insertion to create the discontinuity surface. Finally, h-refinement can be applied along any or all directions to have a refined model which is analysis suitable. Listing 2 is then used to build the element connectivity array for the interface elements.

Listing 7: Matlab code to build NURBS geometry of the 3D DCB

| uKnot = [0 0 1 1]; |

Movies of these analyses can be found at http://www.frontiersin.org/people/NguyenPhu/94150/video.
Figure 22: L-shaped specimen with one initial crack and cohesive elements at all ply interfaces: multiple delaminations.

Figure 23: Three dimensional double cantilever beam: geometry and loading data.
6.3.2. Analysis results

We use an isotropic material with Young modulus $E = 2.1 \times 10^5$ MPa and Poisson ratio $\nu = 0.3$. The material constants for the cohesive law are $G_{IC} = 0.28$ N/mm, $\tau_0^\parallel = 27$ MPa, $K = 10^7$ N/mm$^3$. Two layers of elements are placed along the thickness and the width direction. Figure 24 shows the deformed shape and the load-displacement curve including a comparison with the classical beam theory solution.

![Figure 24: Three dimensional double cantilever beam: contour plot of the transverse stress on the deformed shape (magnification factor of 3).](image)

6.3.3. Analysis results with shell elements

Next, the problem is solved using isogeometric shell elements. We refer to, for instance, [26, 27, 28] for details on isogeometric shell elements. In this section, we adopt the rotation-free Kirchhoff-Love thin shell as presented in [27]. Due to its high smoothness, NURBS are suitable for constructing $C^1$ shell elements without rotation degrees of freedom. In order to fix the rotation at the right end of the beam, we fix the displacements (all components) of the last two rows of control points, see Fig. (25a) and we refer to [27, 44] for details. For each ply is represented by its own NURBS surface, there is automatically a discontinuity between their interface. Therefore, generation of interface elements in this context is straightforward. Each ply is discretized by a mesh of $264 \times 1$ bi-quadratic elements. The number of nodes/control points is 1596. Figure 25 gives the contour plot of damage and the load-displacement curves.

6.4. Singly curved thick-walled laminate

As a 3D example with more complex geometry, we consider a singly curved thick-walled laminate which was studied in [63, 64]. Air-intakes of formula race cars and strongly curved regions of ship hulls provide examples for such thick-walled curved laminates designs. The geometry of the sample is given in Fig. (26). Since the geometry representation
of the object of interest is the same in both CAD and FEA environment, it is very straightforward and fast to get an analysis-suitable model when changes are made to the CAD model, for instance changing the thickness \( t \). This is in sharp contrary to Lagrange finite elements which uses a different geometry representation. This example also shows how a trivariate NURBS representation of a curved thick/thin-walled laminate can be built given a NURBS curve or surface. For computation, the material constants given in Table 2 are used of which the material properties of the plies are taken from [63]. The material constants for the interfaces are not provided in [63], the ones used here are therefore only for computation purposes.

<table>
<thead>
<tr>
<th>( E_{11} )</th>
<th>( E_{22} = E_{33} )</th>
<th>( G_{12} = G_{13} )</th>
<th>( \nu_{12} = \nu_{13} )</th>
<th>( \nu_{23} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>110 GPa</td>
<td>10 GPa</td>
<td>5.00 GPa</td>
<td>0.27</td>
<td>0.30</td>
</tr>
</tbody>
</table>

\( G_{1c} \) \( \Gamma_{1}^{0} \) \( \Gamma_{3}^{0} \) \( \mu \)

<table>
<thead>
<tr>
<th>( G_{1c} )</th>
<th>( G_{11c} )</th>
<th>( \Gamma_{1}^{0} )</th>
<th>( \Gamma_{3}^{0} )</th>
<th>( \mu )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.075 N/mm</td>
<td>0.547 N/mm</td>
<td>80.0 MPa</td>
<td>90.0 MPa</td>
<td>1.75</td>
</tr>
</tbody>
</table>

Table 2: Singly curved thick-walled laminate: material properties.

6.4.1. Geometry and mesh

The geometry of the singly curved thick-walled laminates can be built by first creating a NURBS curve as shown in Fig. (27). Next, an offset of this curve with offset distance \( t \) is created using the algorithm described in [65] which ensures the offset curve has the same parametrization as its base. This allows a tensor-product surface bounded by these two curves can be constructed. Having these two curves, a B-spline surface can be constructed. Knot insertion was then made to build \( C^{-1} \) lines at the ply interfaces. Finally, the cross section is extruded along the width direction.

Figure 25: Three dimensional double cantilever beam modeled by shell elements: (a) B-spline mesh of one ply with fixed control points and (b) contour plot of damage on the deformed shape (magnification factor of 3) and (c) load-displacement curves.
Figure 26: Singly curved thick-walled laminates: geometry configuration. The thickness $t$ is constant.

We refer to Listing 8 for the Matlab code that produces the geometry. Again, the number of ply can be easily changed and interface elements can be placed along any ply interface. NURBS meshes of the sample are given in Fig. (28).

Listing 8: Matlab code to build NURBS geometry of the singly curved thick-walled laminates.

```matlab
1 l = 80; L = 100; t = 10; w = 40; h = 30;
2 no = 4; % number of layers
3 uKnot = [0 0 0 1 1 2 3 4 5 6 6 7 7 7];
4 controlPts = zeros(4,11);
5 controlPts(1:2,1) = [0;0];
6 controlPts(1:2,2) = [0.5*l;0];
7 controlPts(1:2,3) = [1;0];
8 controlPts(1:2,4) = [1+10;0];
9 controlPts(1:2,5) = [1+0.5*L-8;h-3];
10 controlPts(1:2,6) = [1+0.5*L;h];
11 controlPts(1:2,7) = [1+0.5*L+8;h-3];
12 controlPts(1:2,8) = [1+L-10;0];
13 controlPts(1:2,9) = [1+L;0];
14 controlPts(1:2,10) = [1.5*L+1;0];
15 controlPts(1:2,11) = [2*L;0];
16 controlPts(4,:,:,:) = 1;
17 curve = nrbmak(controlPts,uKnot);
18 [oCurve,offsetPts] = offsetCurve(curve,t,alpha,beta,epsilon,maxIter); % offset curve
19 surf = surfaceFromTwoCurves(curve, oCurve);
20 surf = nrbdegelev(surf,[0 1]); % evaluate order => bi–quadratic
21 % h–refinement in Y direction to make sure it is C'{-1} along delamination path
22 knots = [ ];
23 for ip=1:no-1
24    dd = ip/4;
25    knots = [knots dd dd dd ];
26 end
27 surf = nrbkntins(surf,[[] knots ]); % make the solid
28 solid = nrbextrude(surf,[0,0,w]); % make the solid
```
Figure 27: Singly curved thick-walled laminates: building the cross section as a B-spline surface made of the base curve and its offset. The red points denote the control points. It should be emphasized that the offset curve cannot be built simply using an extrusion operation on the base curve.

Figure 28: Singly curved thick-walled laminates: NURBS meshes.
6.4.2. 2D analyses

Since the straight specimen ends were placed in the clamps of a closed-loop controlled servo-hydraulic testing machine [64], in the FE model, the straight ends are not included. The sample is subjected to a compressive force on the right end and fixed in the left end. The laminate is composed of 45 unidirectional (0°) plies of carbon fiber reinforced plastic. The mesh was consisted of 40 × 45 quartic-quadratic NURBS elements and 1760 quartic interface elements. The number of nodes is 7 020 hence the number of unknowns is 14 040. A plane strain assumption is used. The analysis was performed in 121 load increments and the computation time was 1300s on an Intel Core i7 2.8GHz laptop. The delamination pattern and the load-displacement is given in Fig. (29). Note that no effort was made to compare the obtained result with the test given in [64] since it is beyond the scope of this paper.

![Figure 29: Singly curved thick-walled laminate under compression: delamination pattern (left) and load-displacement curve (right).](image)

6.4.3. 3D analyses

For 3D computations, we assume that laminate is composed of 10 unidirectional (0°) plies of carbon fiber reinforced plastic to reduce the computational expense. Furthermore, interface elements are placed only between ply 5 and 6. Along the thickness, only one layer of elements is used for each ply. This might be not sufficient for analysis.

6.5. Some other models

For completeness, in this section we apply the presented isogeometric framework to other commonly encountered composite structures. In Fig. (31c), a glare panel with a circular initial delamination is given (one quarter of the panel is shown due to symmetry) [66]. The NURBS representation of the panel is given in Fig. (31a) in which the coarsest mesh that consists of 2 × 2 quadratic-linear NURBS elements can capture exactly the circle geometry and Fig. (31b) shows a refined mesh. Then, interface elements can be straightforwardly inserted and delamination analyses can be performed Fig. (31c,d). It should be emphasized that the chosen NURBS parametrization given in Fig. (31a) is not unique and there are singular points at the bottom left and top right corners (this, however, does not affect the analysis since no integration points are placed there).

Next, we present a NURBS mesh for the open hole laminate as shown in Fig. (32). The whole sample can be represented by six NURBS patches of which four patches are for the central part. In this figure, we give a parametrization that results in a so-called compatible multi-patch model. Note that across the patch interface, the basis is only $C^0$. Interface elements are generated for each patch independently using the presented algorithm. It should be emphasized that joining NURBS patches of different parametrizations provides more flexibility albeit a non-trivial task. T-splines can be used as a remedy, see e.g., [67].

Finally, treatment of doubly curved composite panels is by an example given in Fig. (33). Using a CAD software, the panel is usually a NURBS surface, Fig. (33)–left, a trivariate representation is constructed using the ideas recently reported in [65], Fig. (33)–middle, and FE analyses can be performed, see Fig. (33)–right.
Figure 30: Singly curved thick-walled laminate under compression: 3D computation.

Figure 31: Glare with a circular initial delamination: (a) NURBS surface with control points and mesh (4 elements), (b) refined mesh, (c,d) deformed shape and damage plot.
Figure 32: Open hole laminate.

Figure 33: Doubly curved composite laminate: from bi-quadratic NURBS surface to trivariate NURBS solid that is suitable for analysis.
7. Conclusion

An isogeometric computational framework was presented for modeling delamination of two and three dimensional composite laminates. By using the isogeometric concept in which the NURBS representation of the composite laminates is maintained in a finite element environment, it was shown by several examples that the time being spent on preparing analysis suitable models for delamination analyses can be dramatically reduced. This fact is beneficial to designing composite laminates in which various geometry parameters need to be varied. The preprocessing algorithms were explained in details and implemented in MIGFEM—an open source code which is available at https://sourceforge.net/projects/cmcodes/. From the analysis perspective, the ultra smooth high order NURBS basis functions are able to produce highly accurate stress fields which is very important in fracture modeling. The consequence is that relatively coarse meshes (compared to meshes of lower order elements) can be adopted and thus the computational expense is reduced.

Although an elaborated isogeometric computational framework was presented for modeling delaminated composites and several geometry models were addressed, there are certainly a certain number of geometries that has not been treated. One example is three dimensional curved composite panels with cutouts. Possibilities for these problems are trimmed NURBS or T-splines for conforming mesh methods and immersed boundary methods for non-conforming techniques. Performance of isogeometric cohesive elements under dynamic loadings and parallelization of NURBS-based finite elements are, among other things, topics of future works.

Acknowledgements

The authors would like to acknowledge the partial financial support of the Framework Programme 7 Initial Training Network Funding under grant number 289361 “Integrating Numerical Simulation and Geometric Design Technology”. Stéphane Bordas also thanks partial funding for his time provided by 1) the EPSRC under grant EP/G042705/1 Increased Reliability for Industrially Relevant Automatic Crack Growth Simulation with the eXtended Finite Element Method and 2) the European Research Council Starting Independent Research Grant (ERC Stg grant agreement No. 279578) entitled “Towards real time multiscale simulation of cutting in non-linear materials with applications to surgical simulation and computer guided surgery”. The authors would like to express the gratitude towards Drs. Erik Jan Lingen and Martijn Stroeven at the Dynaflo Research Group, Houtsingel 95, 2719 EB Zoetermeer, The Netherlands for providing us the numerical toolkit jem/jive.

References


36


