



AN OOP APPROACH FOR AN ISOGEOMETRIC COHESIVE INTERFACE ELEMENT

Edson Moreira Dantas Jr.

Elias Saraiva Barroso

edsonmdantas@yahoo.com.br

Centro Universitário Unichristus

Campus Dom Luis, 60160-230, Ceará, Fortaleza, Brazil

elias.barroso@gmail.com

Universidade Federal do Ceará

Departamento de Engenharia Estrutura e Construção Civil, 60455-760, Fortaleza, Ceará, Brazil

Evandro Parente Junior

evandro@ufc.br

Universidade Federal do Ceará

Departamento de Engenharia Estrutura e Construção Civil, 60455-760, Fortaleza, Ceará, Brazil

Abstract. *Delamination is one of the biggest problems in fracture mechanics. One of the most used method to predict failure is the finite element method though cohesive elements. Since isogeometric methods are obtaining great visibility because of it geometric integration with CAD software (using it proper geometry), the present work discusses the computer simulation of fracture using isogeometric interface element. The discontinuity is inserted through knot insertion. An OOP discussion is raised since the IGA interface element is obtained simply and easily through a FEM /IGA framework. This approach allows the handling of 2D and 3D models with different element shapes and constitutive models in a simple and easily extensible way. The structures geometries are modeled using plane NURBS elements. The computational implementation developed in this work was verified and validated using results available in the literature.*

Keywords: *Fracture Mechanics, Isogeometric Analysis, Delamination, OOP*

INTRODUCTION

Fracture mechanics, through decades, has become an important mathematic tool on crack propagation studies. Computational methods have been developed and improved looking for more realistic and robust solution methods. The computational implementation of these methods are widespread through the use of interface finite element (Xu & Needleman, 1994; Camacho & Ortiz, 1996; Alfano & Crisfield, 2001; Camanho et. al., 2003; Goyal et. al, 2004; Balzani & Wagner, 2008) using the cohesive zone models.

These models consider that the stress transferring between cracks doesn't cease after the crack beginning, but it reduces gradually until zero for a complete opening. This approach is based on constitutive models relating crack surface tractions with crack surface displacement (displacement jump), where the area above this relations is the energy release rate (G).

Many constitutive models have been proposed for modeling this problem. One of the main ideas is the different relation between opening modes (Mode I and Mixed Mode II e III). Besides finite element robustness, a geometry representation flaw is observed on classic finite element formulations. On finite element analysis, a geometry approximation is obtained through polynomial interpolation functions, generating a loss of information during CAD geometry approximation for structural analysis.

For a better precision and CAD-CAE integration, the idea of using CAD design technology (CAD functions) such as B-splines, NURBS (Non-Uniform Rational B-splines) and T-splines as basis functions finite element (FE) framework has become widely accepted on many simulation fields. Isogeometric analysis (IGA) is a isoparametric formulation, based on using NURBS functions to define the domain and the fields of the problem.

According to Nguyen et al. (2013), since the remarkable paper from Hughes et al. (2005), some works, including a detailed monograph, were published entirely on the subject (Cottrell et al., 2009), and many applications were found in structural mechanics, solid and fluid mechanics and contact mechanics. Nguyen et al. (2013) still emphasize that the idea of using CAD technologies in finite elements dates back at least to Kagan et al. (1998) where B-splines were used as shape functions in FEM and subdivision surfaces were adopted to model shells (Cirak et al., 2000).

IGA is also a powerful tool for fracture mechanics analysis. It has been applied to dynamic fracture modeling (Babuška & Melenk, 1997; Luycker et al., 2011) and adaptative refinement for 3D fracture simulation Borden et al. (2011).

An IGA framework was presented on Verhoosel et al. (2011). The high order continuity of NURBS curves was taken advantages for modeling composite (laminated/sandwich) structures for a smooth continuity of stress fields (Verhoosel et al., 2011; Kapoor & Kapania, 2012; Guo et al., 2014; Barroso et al., 2015). Recently Kapoor & Kapania (2012) simulated delamination of composite structures successfully using B-spline as basis function.

This work presents an OOP framework for implementing interface isogeometric element according to a formulation presented on Nguyen et al. (2013) and comments about the advantages of implementing such method on an OOP finite element frameworks. This approach allows the handling of 2D and 3D models with different element shapes and constitutive models in a simple and easily extensible way.

A discussion about NURBS geometry, its basis and operations is raised on Chapter 2. Chapter 3 shows interface isogeometric element formulation. Chapter 5 presents the implementations performed on a C++ code. An overall view of the program classes, its modifications and class creations are commented just like the different cohesive models used. Chapter 6 shows numerical examples and applications of the method. Chapter 7 concludes the advantages of implementing such methods and suggests future works. Acknowledgements and references are presented on the last chapters. The computational implementation developed in this work was verified and validated using results available in literature.

NURBS

Modeling curves and surfaces are possible through implicit and parametric equations. Some forms of representing parametric equations are using Bézier, B-spline and NURBS curves.

NURBS is a parametric representation widely used in computational modeling. Those functions offer possibility of representing mathematical models (quadratic, conical, revolution surfaces, . . .) and free form models using the same data base for storage both. A C NURBS of degree p is defined as a linear combination between control points $\mathbf{p}_i$ and functions of rational base $R_{i,p}(\xi)$, as shown on expression:

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

where ξ is the parametric coordinate of curve evaluation and n is the number of basis. The NURBS basis are evaluated in function of B-Spline $N_{i,p}$ and weights w_i given by the expression:

$$R_i(\xi) = \frac{N_{i,p} w_i}{\sum_{\hat{i}=1}^n N_{\hat{i},p} w_{\hat{i}}} = \frac{N_{i,p} w_i}{W(\xi)} \quad (2)$$

where $W(\xi) = \sum_{\hat{i}=1}^n N_{\hat{i},p}(\xi) w_{\hat{i}}$ is the weight function. Each control point $\mathbf{p}_i$ has a corresponding weight w_i . The use of weights permits changing the final geometry of the model and modeling conical curves (circumferences and ellipsis).

B-splines are curves capable of describing several distinct segments along the same parametric representation. Those attributes are obtained limiting the base function actions on parametric space regions. Those regions are known as knot spans and are defined by a vector of parametric values, non-null and non-decreasing values. Limited on the parametric interval $[\xi_1, \xi_{n+p+1}]$ on which the curve was defined.

Considering the knot vector $\Xi = [\xi_1, \xi_2, \dots, \xi_{n+p+1}]$, the B-Spline base functions defined by the recursive formula Cox-de Boor (Piegl & Tiller, 1997):

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

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \quad (3)$$

Each $N_{i,p}$ basis contributes on a parametric interval $[\xi_i, \xi_{i+p+1}]$. The number of bases n might be defined in function of the knot vector size ks and degree p as below:

$$n = ks - p - 1 \quad (4)$$

The derivative $N_{i,p}$ might be calculated as:

$$\frac{d}{d\xi} N_{i,p}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \quad (5)$$

B-Spline functions $N_{i,p}(\xi)$ have the following features:

- Non negative: $N_{i,p}(\xi) \geq 0$;
- Partition of Unit properties: $\sum_{i=1}^n N_{i,p}(\xi) = 1$;
- Localized supports: $N_{i,p}(\xi) = 0$ if ξ is out of the interval $[\xi_i, \xi_{i+p+1}]$;
- Continuity C^{p-1} between *knots*;
- Continuity C^{p-m} on *knots*, where m is the knot multiplicity.

Another important observation is that the number of bases that contributes for each parametric interval (knot span) is ways equals to $p + 1$. It is important to notice that rational base functions $R_{i,p}(\xi)$ heirsch the B-Spline basis properties $N_{i,p}(\xi)$.

A solid NURBS defined by a tensorial product is built by the product of three univariant NURBS base functions. Given a tensor of control points P ($n \times m \times o$), a NURBS of degree p on direction ξ with a knot vector $\Xi = [\xi_1, \xi_1, \dots, \xi_{n+p+1}]$, a NURBS of degree q on direction η with a knot vector $\mathcal{H} = [\eta_1, \eta_1, \dots, \eta_{m+q+1}]$ and a NURBS of degree l on direction ζ with a knot vector $\mathcal{L} = [\zeta_1, \zeta_1, \dots, \zeta_{o+l+1}]$, defines a solid V as:

$$V(\xi, \eta, \zeta) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^o R(\xi, \eta, \zeta)_{ijk} \mathbf{P}_{ijk} \quad (6)$$

where $R(\xi, \eta, \zeta)$ is the trivariate base function given by:

$$R(\xi, \eta, \zeta)_{i,j,k} = \frac{w_{ijk} N_{i,p}(\xi) N_{j,q}(\eta) N_{k,l}(\zeta)}{W(\xi, \eta, \zeta)} \quad (7)$$

and $W(\xi, \eta, \zeta)$ is the trivariate weight function given by:

$$W(\xi, \eta, \zeta) = \sum_{\hat{i}=1}^n \sum_{\hat{j}=1}^m \sum_{\hat{k}=1}^o w_{\hat{i}\hat{j}\hat{k}} N_{\hat{i},p}(\xi) N_{\hat{j},q}(\eta) N_{\hat{k},l}(\zeta) \quad (8)$$

Partial derivatives $W(\xi, \eta, \zeta)$ are evaluated by the expressions:

$$\begin{aligned} \frac{\partial}{\partial \xi} W(\xi, \eta, \zeta) &= \sum_{\hat{i}=1}^n \sum_{\hat{j}=1}^m \sum_{\hat{k}=1}^o w_{\hat{i}\hat{j}\hat{k}} N'_{\hat{i},p}(\xi) N_{\hat{j},q}(\eta) N_{\hat{k},l}(\zeta) \\ \frac{\partial}{\partial \eta} W(\xi, \eta, \zeta) &= \sum_{\hat{i}=1}^n \sum_{\hat{j}=1}^m \sum_{\hat{k}=1}^o w_{\hat{i}\hat{j}\hat{k}} N_{\hat{i},p}(\xi) N'_{\hat{j},q}(\eta) N_{\hat{k},l}(\zeta) \\ \frac{\partial}{\partial \zeta} W(\xi, \eta, \zeta) &= \sum_{\hat{i}=1}^n \sum_{\hat{j}=1}^m \sum_{\hat{k}=1}^o w_{\hat{i}\hat{j}\hat{k}} N_{\hat{i},p}(\xi) N_{\hat{j},q}(\eta) N'_{\hat{k},l}(\zeta) \end{aligned} \quad (9)$$

Partial derivatives of the trivariate base functions is defined by:

$$\begin{aligned}
 \frac{\partial}{\partial \xi} R_{i,j,k}(\xi, \eta, \zeta) &= w_{ijk} N_{j,q}(\eta) N_{k,l}(\zeta) \frac{W(\xi, \eta, \zeta) N'_{i,p}(\xi) - \frac{\partial}{\partial \xi} W(\xi, \eta, \zeta) N_{i,p}(\xi)}{W^2(\xi, \eta, \zeta)} \\
 \frac{\partial}{\partial \eta} R_{i,j,k}(\xi, \eta, \zeta) &= w_{ijk} N_{i,p}(\xi) N_{k,l}(\zeta) \frac{W(\xi, \eta, \zeta) N'_{j,q}(\eta) - \frac{\partial}{\partial \eta} W(\xi, \eta, \zeta) N_{j,q}(\eta)}{W^2(\xi, \eta, \zeta)} \\
 \frac{\partial}{\partial \zeta} R_{i,j,k}(\xi, \eta, \zeta) &= w_{ijk} N_{i,p}(\xi) N_{j,q}(\eta) \frac{W(\xi, \eta, \zeta) N'_{k,l}(\zeta) - \frac{\partial}{\partial \zeta} W(\xi, \eta, \zeta) N_{k,l}(\zeta)}{W^2(\xi, \eta, \zeta)}
 \end{aligned} \quad (10)$$

A NURBS model might be refined adding new knots values or elevating the function degree, without changing the final geometry. *Knot Insertion* and *degree elevation* (Piegl & Tiller, 1997) The *Knot Insertion* operation is an operation performed on B-splines used to change the curve description without changing geometry. A knot insertion adds a new value of ξ_i on the knot vector Ξ , a new base $N_{i,p}$ and a new control point. For maintaining the curve identical, some control points are changed. Considering a knot vector $\Xi = [\xi_1, \xi_2, \dots, \xi_{n+p+1}]$ and a parametric value to be inserted $\bar{\xi} \in [\xi_k, \xi_{k+1}]$, the new control points $\bar{\mathbf{p}}$ might be obtained in function of the primary control points $\mathbf{p}$ by the expression:

$$\bar{\mathbf{p}}_i = \begin{cases} \mathbf{p}_1, & i = 1, \\ \alpha_i \mathbf{p}_i + (1 - \alpha_i) \mathbf{p}_{i-1}, & 1 < i < h, \\ \mathbf{p}_{h-1} & i = h. \end{cases} \quad (11)$$

where h is the size of the new control point vector $\bar{\mathbf{p}}$. The term α is calculated by:

$$\alpha_i = \begin{cases} 1, & 1 \leq i \leq k - p, \\ \frac{\bar{\xi} - \xi_i}{\xi_{i+p} - \xi_i}, & k - p + 1 \leq i \leq k, \\ 0, & i \geq k + 1. \end{cases} \quad (12)$$

A knot insertion adds a new value of ξ_i on the knot vector Ξ , a new base $N_{i,p}$ and a new control point. Once inserted knots with a multiplicity $m = p$, the continuity of the curve is disturbed (forming a cusp) isolating the modifications performed on a parametric interval from another. From an IGA analysis, knot insertion has a fundamental importance on discretization of the model. For interface IGA, knot insertion is important on discontinuity element insertion.

ISOGEOMETRIC INTERFACE ELEMENT FORMULATION

The following section describes the formulation of zero thickness interface elements in the context of Isogeometric Analysis.

Interface Finite Element Formulation

Interface finite elements are a fancy way to describe discontinuity inside a continuous body using continuous finite elements. This formulating, according to Virtual Work Principles requires a division of a whole body into two sub-domains and both of them must be in equilibrium ((Goyal et. al, 2004)).

$$W_{int} + W_{cohe} = W_{ext} \quad (13)$$

$$\int_V \sigma_{ij} \delta \epsilon_{ij} dV + \int_{\bar{S}} T_i \delta \Delta_i d\bar{S} = \int_V b_i \delta u_i dV + \int_S T_i \delta u_i dS$$

The internal virtual work (W_{int}) is defined as:

$$W_{int} = \delta \mathbf{u}^T \int_V \bar{\mathbf{B}}^T \boldsymbol{\sigma} dV = \delta \mathbf{u}^T \mathbf{g} \quad (14)$$

The structure internal force vector $\mathbf{g}$ is given as:

$$\mathbf{g} = \int_V \bar{\mathbf{B}}^T \boldsymbol{\sigma} dV = \delta \mathbf{u}^T \mathbf{g} \quad (15)$$

The external virtual work (W_{ext}) is defined as:

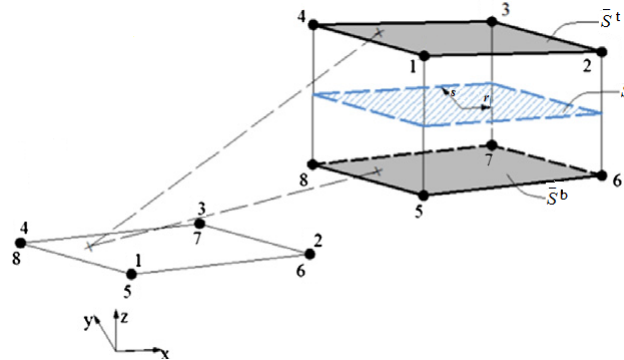
$$W_{ext} = \delta \mathbf{u}^T \int_V \mathbf{N}^T \mathbf{b} dV + \delta \mathbf{u}^T \int_S \mathbf{N}^T \mathbf{q} dS \quad (16)$$

The structure external force vector $\mathbf{f}$ is given as:

$$\mathbf{f} = \int_V \mathbf{N}^T \mathbf{b} dV + \int_s \mathbf{N}^T \mathbf{q} dS \quad (17)$$

For a zero thickness interface finite element, the formulation consist on determine the displacement jump between top and bottom faces.

Figure 1: 3D Interface Element of 8 nodes



The relative displacements are given according Equation 18

$$\Delta = \sum_{i=1}^n (N_i u_i^t - N_i u_i^b) = \mathbf{B} \mathbf{u} \quad (18)$$

where $\mathbf{u}$ is the nodal displacement vector on global coordinate system, N_i represents the shape functions of the element, n represents the number of control points on each surface of the element and $\mathbf{B}$ represents the strain-displacement matrix, given by Equation 19:

$$\mathbf{B} = \begin{bmatrix} N_i & 0 & 0 & \dots & | & -N_i & 0 & 0 & \dots \\ 0 & N_i & 0 & \dots & | & 0 & -N_i & 0 & \dots \\ 0 & 0 & N_i & \dots & | & 0 & 0 & -N_i & \dots \end{bmatrix} \quad (19)$$

A surface reference $\bar{S}$ illustrated in Fig. 1 is defined as a middle surface between superior and inferior element surface:

$$\mathbf{x} = \sum_{i=1}^n \frac{N_i \mathbf{x}_i^t + N_i \mathbf{x}_i^b}{2} \quad (20)$$

Initially, the tangent vectors are computed as:

$$\mathbf{a}_1 = \mathbf{x}_{,r}; \quad \mathbf{a}_2 = \mathbf{x}_{,s} \quad (21)$$

The normal vector is calculated as:

$$\mathbf{a}_3 = \mathbf{a}_1 \times \mathbf{a}_2 \quad (22)$$

However the vector $\mathbf{a}_1$ and $\mathbf{a}_2$ are not always perpendicular. For defining a better a orthogonal base, vector $\mathbf{a}_1$ is defined as the first tangential vector of the local system , while $\mathbf{a}_3$ is used to calculate the normal vector system:

$$\mathbf{e}_1 = \frac{\mathbf{a}_1}{|\mathbf{a}_1|}; \quad \mathbf{e}_3 = \frac{\mathbf{a}_3}{|\mathbf{a}_3|} \quad (23)$$

Finally, the second tangential vector of the local system is defined as:

$$\mathbf{e}_2 = \mathbf{e}_3 \times \mathbf{e}_1 \quad (24)$$

The displacement transformation from global system (Δ) to local system (δ) is defined by the following relation:

$$\delta = \mathbf{R} \Delta \quad (25)$$

where the rotation matrix $\mathbf{R}$ is given by:

$$\mathbf{R} = \begin{bmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{bmatrix} \quad (26)$$

where:

$$\mathbf{e}_1 = \begin{Bmatrix} l_1 & m_1 & n_1 \end{Bmatrix}; \quad \mathbf{e}_2 = \begin{Bmatrix} l_2 & m_2 & n_2 \end{Bmatrix}; \quad \mathbf{e}_3 = \begin{Bmatrix} l_3 & m_3 & n_3 \end{Bmatrix} \quad (27)$$

Through Virtual Work Principles it is possible to show the transformation of traction from local to global system follows:

$$\mathbf{T} = \mathbf{R}^T \mathbf{t} \quad (28)$$

where $\mathbf{t}$ are the cohesive tractions on local system. The internal force vector $\mathbf{g}_{cohe}$ of the interface element is obtained through Virtual Principal Work.

$$\delta W_{cohe} = \delta \mathbf{u}^T \mathbf{g} = \int_{\bar{S}} \delta \Delta^T \mathbf{T} d\bar{S} \quad (29)$$

Using the Equation (18) and Equation (28), the internal force vector is described as:

$$\mathbf{g} = \int_{\bar{S}} \mathbf{B}^T \mathbf{R}^T \mathbf{t} |\mathbf{J}| d\bar{S} \quad (30)$$

where $\bar{S}$ corresponding the element middle surface

The tangent matrix ($\mathbf{K}_T$) is obtained by the internal forces of the element related to the nodal displacements:

$$\mathbf{K}_T = \frac{\partial \mathbf{g}}{\partial \mathbf{u}} = \int_{\bar{S}} \mathbf{B}^T \mathbf{R}^T \frac{\partial \mathbf{t}}{\partial \boldsymbol{\delta}} \frac{\partial \boldsymbol{\delta}}{\partial \boldsymbol{\Delta}} \frac{\partial \boldsymbol{\Delta}}{\partial \mathbf{u}} d\bar{S} = \int_{\bar{S}} \mathbf{B}^T \mathbf{R}^T \mathbf{C}_T \mathbf{R} \mathbf{B} d\bar{S} \quad (31)$$

where:

$$\mathbf{C}_T = \partial \mathbf{t} / \partial \boldsymbol{\delta} \quad (32)$$

is the constitutive tangent matrix of the model.

Isogeometric Analysis

For determining the displacement and deformation field of a structural problem, an isogeometric analysis considers the geometry and displacement fields of a body as a linear combination of np functions of base $\mathbf{R}$ and control points $p_a = (x_a, y_a, z_a)$, expressed as Equation 33:

$$x = \sum_{a=1}^{np} R_a x_a; \quad y = \sum_{a=1}^{np} R_a y_a; \quad z = \sum_{a=1}^{np} R_a z_a \quad (33)$$

Considering a geometry being described by NURBS, the R_a functions are rational base functions $R_{i,j,k}(\xi, \eta, \zeta)$ represented on a parametric space. Using those shape functions, it is possible to use the classical isoparametric finite element formulations with some concepts adoptions.

The first concept is the physical idea of an element. On finite element method, the concept of element is defined by a set of nodes (points on the geometry) used to define a sub-domain of the geometry. On isogeometric analysis these sub domains are not only defined by points outside the geometry (control points) but also defined by parametric intervals (knot spans). The knot spans are generally represented by a vector containing values varying from 0 to 1. These values are called knots. Each knot might be associated to one or more IGA elements. Once NURBS function has only local support, only $p + 1$ basis functions are not null on an knot span, as illustrated in Fig. 2.

NURBS might be used to describe entire geometries. Some times, on complex geometries, might be necessary to define geometries using more than one NURBS geometry. A geometry defined by a single NURBS is known as patch.

On this way, although NURBS functions are defined on entire patch, the stiffness matrix are calculated using Gauss quadrature on each knot span (each element) and summed to global stiffness matrix exactly as FEM.

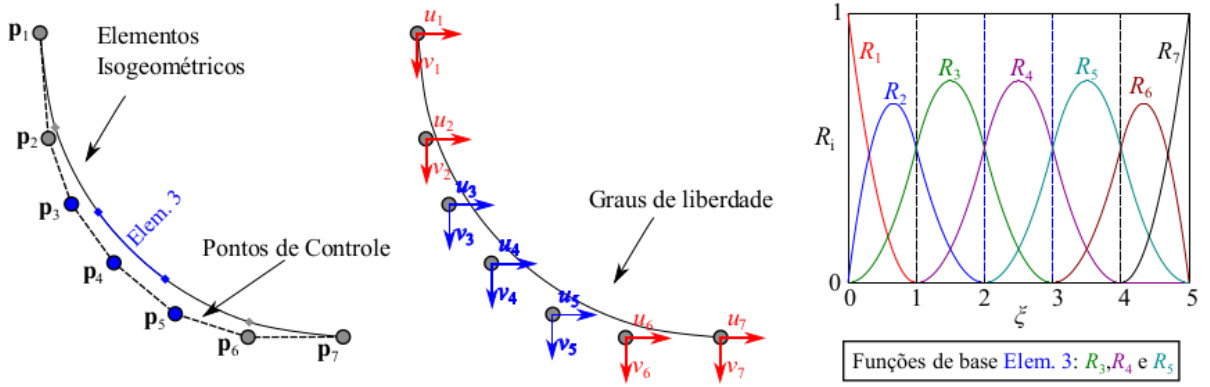


Figure 2: Example of a cable modeled with IGA elements, showing the degrees of freedom and base functions for the third element (blue).

For integration, the element is mapped from physical space to NURBS parametric space and, on sequence, mapped to $[-1,1]$ interval, as illustrated in Fig. 3.

This requires the mapping of Gauss coordinates $(\hat{\xi}, \hat{\eta}, \hat{\zeta})$ to NURBS parametric space (ξ, η, ζ) , by the expression:

$$\xi = \xi_{in} + (\hat{\xi} + 1) \frac{\xi_{fi} - \xi_{in}}{2}; \quad \eta = \eta_{in} + (\hat{\eta} + 1) \frac{\eta_{fi} - \eta_{in}}{2}; \quad \zeta = \zeta_{in} + (\hat{\zeta} + 1) \frac{\zeta_{fi} - \zeta_{in}}{2} \quad (34)$$

where ξ_{in} , ξ_{fi} , η_{in} , η_{fi} , ζ_{in} e ζ_{fi} are the values of knot span elements on directions ξ , η e ζ .

With the Gauss coordinates mapped, it is possible to calculate the base functions R_i and its $(\hat{\xi}, \hat{\eta}, \hat{\zeta})$ derivatives R'_i of the element.

The derivatives should be mapped back to reference element space, being for this necessary to calculate Jacobian transformation with terms similar to $\frac{\partial \xi}{\partial \hat{\xi}} = \frac{\xi_{fi} - \xi_{in}}{2}$. A manner of avoiding the second mapping defined here as using Bézier extraction.

Discontinuity inserction

Since a B-Spline of p degree is at least $p - k$ times differentiable on a knot of multiplicity k , the continuity of such curve is defined by the same rule. Looking for inserting a discontinuity, considers a quadratic B-spline defined using $\Xi = [0, 0, 0, 0.5, 1, 1, 1]$. A B-spline for this knot vector is shown in Fig. 4a) and the basis functions for this curve are shown in Fig. 5a). Changing the multiplicity of knot $\xi = 0.5$ to $p + 1$, ($\Xi = [0, 0, 0, 0.5, 0.5, 0.5, 1, 1, 1]$), a rupture is observed on curve (C^{-1}) continuity. Figure 4b) shows a representation of this discontinuous curve (moved slightly forward). The red marks represent the control points. Figure 5b) shows a representation of the new basis functions. Since knot insertion were performed the control points matrix should be changed. Consider a B-spline curve with the control point matrix defined by $\mathbf{A}$ as shown in Equation 35. The new control point matrix will be defined by $\mathbf{A}'$, also given.

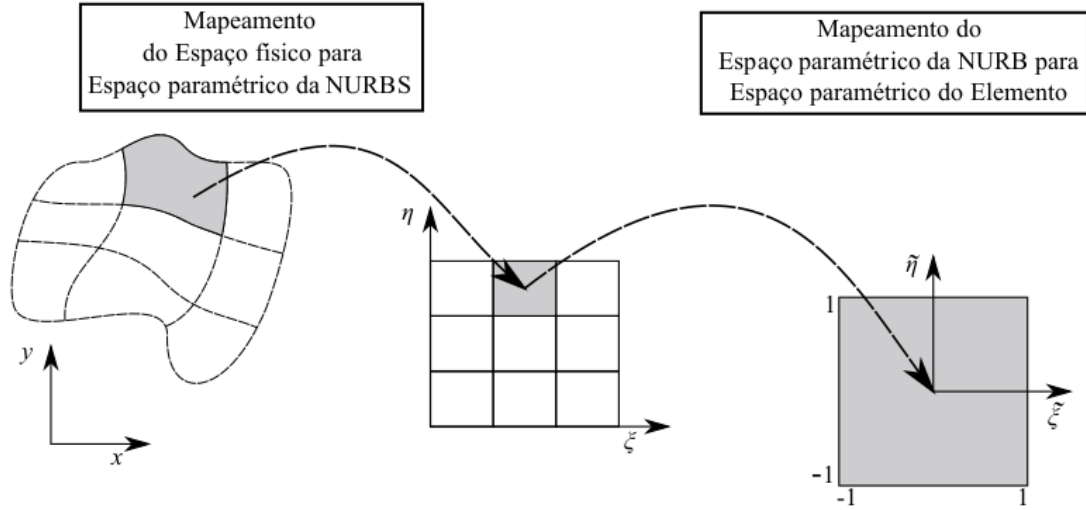
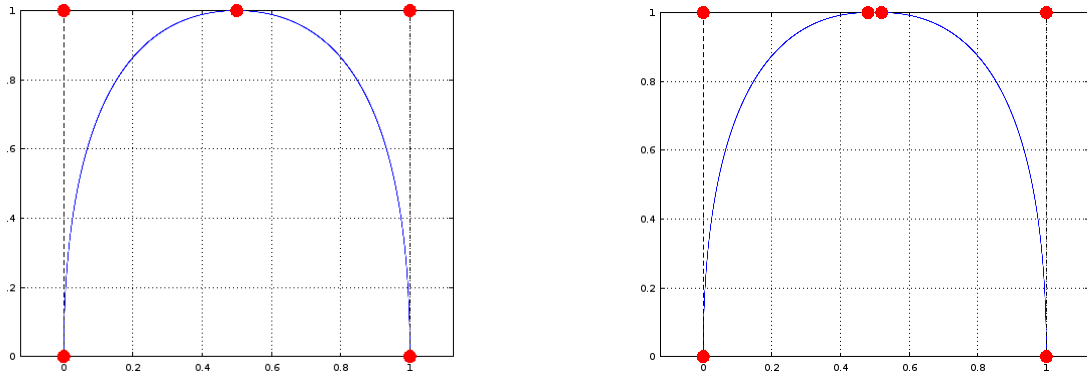


Figure 3: Example of AIG elements showing it degrees of freedom and shape functions.

Figure 4: $p+1$ knot insertion to introduce a $C-1$ discontinuity on B-spline curve at $\xi = 0.5$

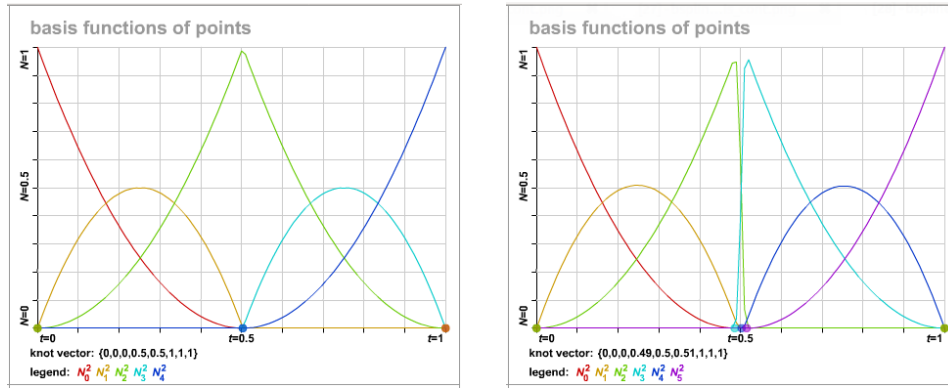


a) Continuous B-spline curve before knot insertion b) Discontinuous B-spline curve after knot insertion

$$\mathbf{A} = \begin{bmatrix} 0.0 & 0.0 \\ 0.0 & 1.0 \\ 0.5 & 1.0 \\ 1.0 & 1.0 \\ 1.0 & 0.0 \end{bmatrix} \quad \mathbf{A}' = \begin{bmatrix} 0.0 & 0.0 \\ 0.0 & 1.0 \\ 0.5 & 1.0 \\ 0.5 & 1.0 \\ 1.0 & 1.0 \\ 1.0 & 0.0 \end{bmatrix} \quad (35)$$

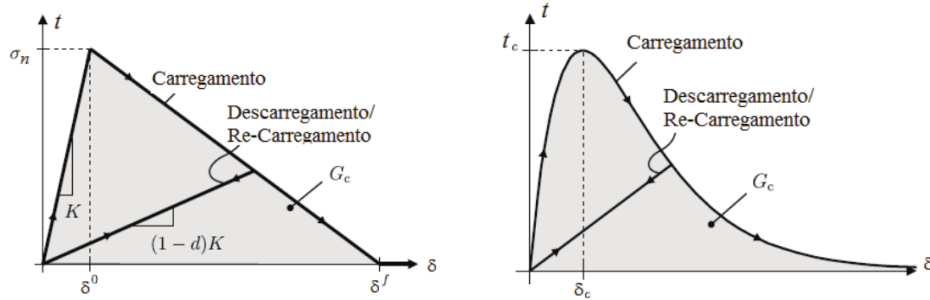
This formulation is defined on the sense of inserting discontinuity inside a patch, defining an interface IGA element between two IGA elements. The parametric definition of a interface element is dependent of direction of interface element insertion. Figure 9 shows a example of interface IGA direction definition for fracture modeling on a rectangular bar subjected to

Figure 5: Basis function before and after knot insertion to introduce a C-1 discontinuity on B-spline curve at $\xi = 0.5$



a) B-Spline Basis for continuous curve. b) B-Spline basis for discontinuous curve.

Figure 6: Constitutive laws.



a) Bilinear law.

b) Exponential law.

traction. The interface IGA element is inserted on η direction, between both IGA elements.

CONSTITUTIVE MODELS

The constitutive models are used for representing the behavior of cohesive cracks. The tension-relative displacement relation might be defined by different laws. Two of the most used relations are the bilinear and exponential laws. Figure 6 show some of these models characteristics. Both models include damage evolution and load/unload behavior:

A bilinear model for delamination and a mixed mode, which the fracture toughness is a phenomenological function, rather than a material constant, of mode mixity as formulated by Kenane & Benzeggagh (1996). This development is presented by Turon (2006) for composite delamination. On this work only bilinear model I law was used. The constitutive law is described by:

$$t = (1 - d)K \delta + dK \langle -\delta \rangle \quad (36)$$

where t is the traction and δ is the displacement jump on element description system. d is the damage variable ($0 \leq d \leq 1$). Dantas Junior (2014) works shows other constitutive laws

including the mixed mode (mode II and mode III) proposed by Kenane & Benzeggagh (1996). where:

$$t = \begin{cases} K\delta & \text{se } \delta_{max} \leq \delta_0 \quad \text{or} \quad \delta < 0 \\ K_s\delta & \text{se } \delta_0 < \delta_{max} < \delta_f \\ 0 & \text{se } \delta_{max} \geq \delta_f \end{cases} \quad (37)$$

where the tangent stiffness K_s is given by:

$$K_s = (1 - d)K \quad (38)$$

The maximum displacement δ_{max} is obtained along the analysis, δ_0 is the displacement limit before the damage beginning (elastic limit) and δ_f is the final displacement before crack opening on a given material point. Those parameters are illustrated in Fig. 6, where:

$$\delta_0 = \frac{\sigma_n}{K}, \quad \delta_f = \frac{2G_{Ic}}{\sigma_n}, \quad (39)$$

G_{Ic} is the critical energy release rate for mode I and σ_n is the traction opening resistance.

OOP IMPLEMENTATION ASPECTS

All the implementations were performed on F.A.S.T., a parallel finite element open source software. All the IGA implementations are based on Bézier extraction concepts. The main idea of using this concept is keeping the software loyal to OOP philosophy, taking advantages of its benefits.

F.A.S.T. architecture

F.A.S.T. is based on C++ programming language. Using the Object Oriented Programming paradigm, implementing new methods and element without changing the overall class structure of the software. The general class structure of FAST can be seen in Fig. 7. In a global level, the Control class is responsible for controlling the analysis process, performing the stiffness matrix assembly, the external load vector evaluation and solving the global equilibrium equations using the Newton-Raphson method if the analysis includes geometric non-linearity. The cElement class handles individual tasks of each element of a mesh and contains: a Shape object, which stores the shape functions (N); an analysis model object (AnModel), responsible for the definition of the degrees of freedom and the displacement and strain fields; and an analysis model for the element section (SecAnalysis), which handles routines such as material coordinate transformations. Also, in the numerical integration of the element matrices, each integration point is an object of the class IntPoint, which stores the parametric coordinates and the weight of each one. The Node class is basically a storage class, containing the nodal coordinates, support conditions and stiffness springs. The Material class stores the material properties and the Load class handles the different kinds of loads which can be applied in a finite element mesh, including the evaluation of the external load vector of each element.

The *cShape* class stores geometric information of the elements. Such information is described from nodal incidences, stored on reference vector of *cNode* class. This class has three

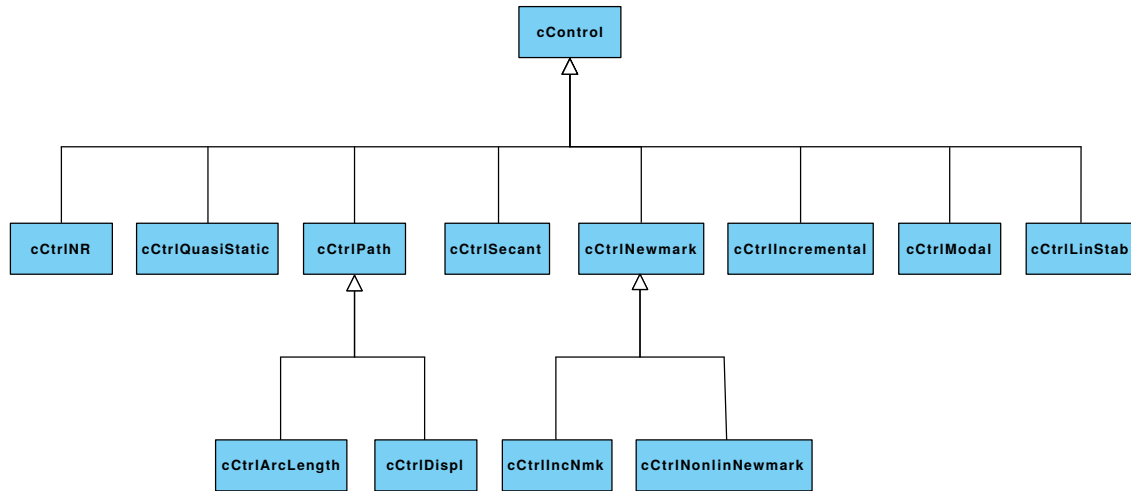


Figure 7: General class structure of FAST.

most important methods: a) ShpFunc(): Responsible for calculating the shape function values of the finite element for a given parametric coordinate. b) DrvShpRST(): Responsible for evaluating the derivatives of shape functions on element parametric space, for a given parametric coordinate. c) DrvShpXYZ(): Responsible for evaluating the derived shape functions on cartesian space, for an especific parametric coordinate. This function evolves the Jacobian determination, needing on this way, to use the DrvShpRST() method on it implementation.

The cShapeIGA class handle the IGA elements features related to the field and geometric (shape) interpolation, as the computation of shape functions and their derivatives. This class, also gives access to the patch object associated to the IGA element. The control points are loaded on the same data structure used to store the finite element nodes. The element incidence is determined based on element knot span, considering the patch topology associated to the element. Figure 8 shows an UML diagram the interaction of this class with other classes, including the new classes that will be presented.

The cPatch class stores data and methods related to a NURBS. Three sub-classes are used to implement and deal with curves, surfaces and solids NURBS. Only the incidences of patch control points are loaded, since those points where instantiated and stored before on cNode. The Evaluate() method evaluates NURBS for a given parametric coordinate. cBsplineBasis deals with necessary data for the B-spline basis function. The cKnotVector stores the knot vector values considered. The B-spline objects are responsables for evaluating and storing the Bézier extraction matrices for each knot span. This task is performed by the LoadCMat() method after the reading of the cBsplineBasis object.

Bézier extaction method allows defining different B-splines basis of different elements, as a linear combination of the same group of Bézier basis, this allows the computational implementation become very close to classic finite element methods. Another advantage of using Bézier extraction method, is the no needing of multiple parametrization from cartesian space to bspline space to Gauss space This is possible because Bézier basis are defined on a single interval making possible to define the same curve on $\Xi[-1, \dots, 1]$ space. Barroso et al. (2015) shows details of Bézier extraction method and it implementation.

It is important to remark that multiple heritage, allowed structuring the software without

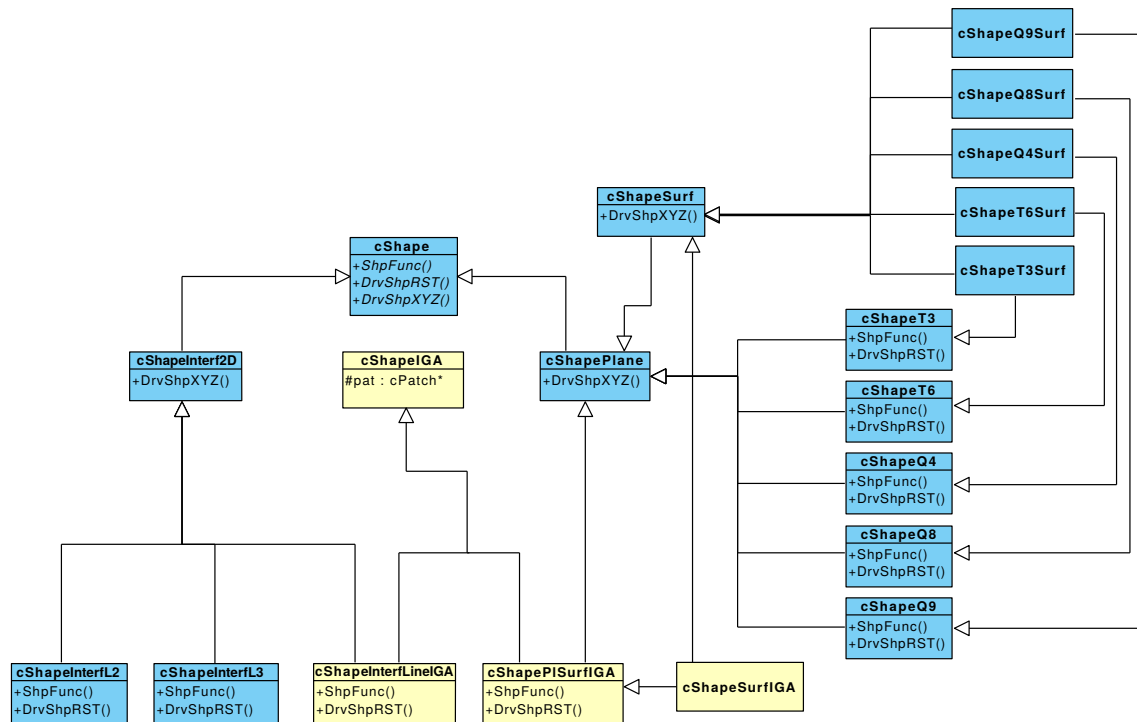


Figure 8: UML diagram of implemented new classes (yellow) and isogeometric analysis class structure (blue).

the needing of reimplementing DrvShpXYZ. Those methods are implemented on a generic way on heirs classes (eg: cShapeSolid, cShapePlane e cShape-Surf).

Based on the same concept of multiple inheritance, a single class were created to manage the interface IGA data. The cShapeInterfLineIGA class is inherited from cShapeIGA and cShapeInterf2D class. All the interface finite element classes were previously implemented on F.A.S.T. Dantas Junior (2014) shows detail of this implementation.

cShapeInterf2D class, manages the different types of line interface elements. This class also computes the Rotation matrix for different types of line interface elements.

When an object of cShapeInterfLineIGA is instantiate, the class initially stores the information about the element direction on patch object. On a sequence, the class computes the Shape Function and it derivatives (DrvShpXYZ) according to the correct control points positioned on the region which the discontinuity is wanted and the knot (element transverse direction) and knot span (element longitudinal direction) of the patch. Figure 8 shows an UML diagram of those classes and it main methods. The numerical integration where performed using Gauss points (Full integration). Other integration points are available on F.A.S.T. as Lobatto points.

NUMERICAL EXAMPLES

In this item are performed a numerical analysis using cohesive zone model. A pure mode I test is perform two verify the implementation.

One Element Test

As the constitutive model is previously validated on Dantas Junior (2014), our concern is limited to IGA interface element validation. A single patch with two continuous 2D IGA elements connected by a interface IGA element were modeled. The base of one of the elements was fixed and the other face is on traction according to Fig. 9.

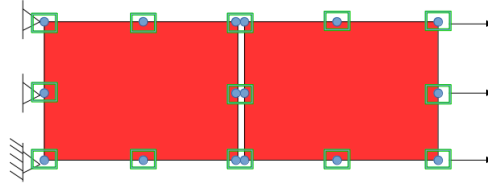


Figure 9: Single patch model with discontinuity: blue marks represents the control points and green marks the knot. Interface IGA is defined by a knot on ξ direction and a knot span on η direction .

The same parameters used on F. Evangelista Jr & Proença (2012) are used on this work. A bar with $L=0.1\text{m}$ and a transversal section of side sized 0.25m . The material properties are described on Table 1 below.

Table 1: Material properties

$E(\text{GPa})$	ν	$\sigma_n(\text{MPa})$	$G_{Ic}(\text{N/m})$	$K_o(\text{Pa/m})$
27.0	0.0	5.0	99.1	5e13

The interface IGA was compared with finite element results:

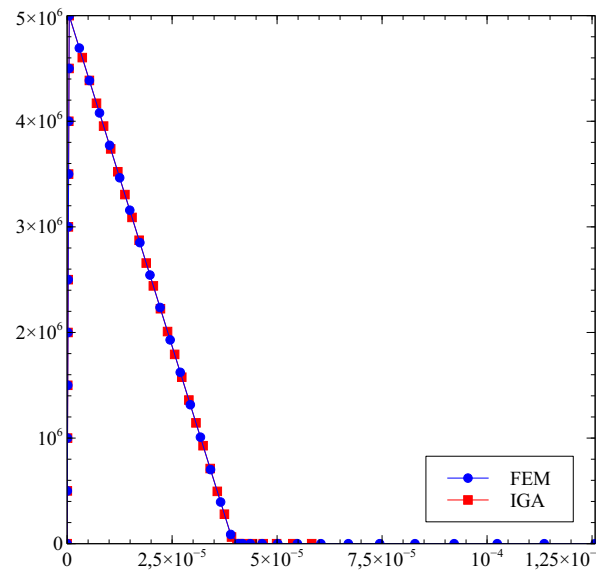


Figure 10: Results obtained in Example 1.

Good results where obtained since it is difficult distinguish the results each method results.

Both presented the softening behavior expected for this case and a maximum traction resistance of 5 MPa.

CONCLUSION

The present work showed the advantages of integrating an OOP finite element framework through implementing an interface element. Although the presented implementation here was based on a line interface element (2D IGA), this implementation is expandable for plane interface elements (3D IGA) on the same fancy way. Since F.A.S.T. has already some parametric elements implemented as 2D/3D laminated elements and many cohesive constitutive models, the structure for receiving this new element was almost perfectly ready. The software, now, is prepared for modeling composite delamination, laminated glass delamination and glare delamination failures problems with IGA elements.

This approach was only possible by using the Bézier extraction idea, where every IGA element is formulated with the same shape functions basis. As a suggestion for future works, other examples must be performed for a better performance evaluation of IGA interface elements over interface finite elements.

ACKNOWLEDGEMENTS

To my wife for her patience and emotional support, Centro Universitário Unichristus for the financial support and to LMCV-UFC lab for the incredible partnership.

REFERENCES

- Alfano & Crisfield 2001. Finite element interface models for the delamination analysis of laminated composites: mechanical and computational issues. *International Journal for Numerical Methods in Engineering* 50:1701–1736.
- Babuška, I. & Melenk, J. M. 1997. The partition of unity method. *International Journal for Numerical Methods in Engineering* 40:727–758.
- Balzani, C. & Wagner, W. 2008. An interface element for the simulation of delamination in unidirectional fiber-reinforced composite laminates. *Engineering Fracture Mechanics* 75:2597 – 2615.
- Barroso, E. S., Junior, E. P., Melo, A. M. C., Junior, E. M. D., & Junior, J. B. M. S. 2015. Análise isogeométrica de estruturas laminadas. *Proceedings of the XXXVI Iberian Latin-American Congress on Computational Methods in Engineering* :-.
- Borden, M. J., Scott, M. A., Evans, J. A., & Hughes, T. J. R. 2011. Isogeometric finite element data structures based on bézier extraction of nurbs. *International Journal for Numerical Methods in Engineering* 87:15–47.
- Camacho & Ortiz 1996. Computational modelling of impact damage in brittle material. *International Journal of Solids and Structures* 33:2899–2938.
- Camanho et. al., Dávila, M. 2003. Numerical simulation of mixed mode progressive delamination in composite materials. *Journal of Composite Materials* 37.

- Cirak, F., Ortiz, M., & Schröder, P. 2000. Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *International Journal for Numerical Methods in Engineering* 12:2039 – 2072. -.
- Cottrell, J. A., Hughes, T. J., & Bazilevs, Y. 2009. Isogeometric Analysis: Toward Integration of CAD and FEA. John Wiley & Sons Ltd.
- Dantas Junior, E. M. 2014. Análise não linear de compósitos laminados utilizando método dos elementos finitos. Dissertação de mestrado, Universidade Federal do Ceará.
- F. Evangelista Jr, J. R. R. & Proença, S. P. 2012. Three-dimensional cohesive zone model for fracture of cementitious materials based on the thermodynamics of irreversible processes. *Engineering Fracture Mechanics* 97:261–280.
- Goyal et. al, Johnson E. R, D. C. G. 2004. Irreversible constitutive law for modeling the delamination process using interfacial surface discontinuities. *Composite Structures* 65:289–305.
- Guo, Y., Nagy, A. P., & Gürdal, Z. 2014. A layerwise theory for laminated composites in the framework of isogeometric analysis. *Composite Structures* 0:447–457.
- Hughes, T., Cottrell, J., & Bazilevs, Y. 2005. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* 194:4135 – 4195.
- Kagan, P., Fischer, A., & Bar-Yoseph, P. Z. 1998. New b-spline finite element approach for geometrical design and mechanical analysis. *International Journal for Numerical Methods in Engineering* 3:435 – 458. -.
- Kapoor, H. & Kapania, R. 2012. Geometrically nonlinear {NURBS} isogeometric finite element analysis of laminated composite plates. *Composite Structures* 94:3434 – 3447.
- Kenane, M. & Benzeggagh, M. L. 1996. Measurement of mixed-mode delamination fracture toughness of unidirectional glass/epoxy composites with mixed-mode bending apparatus. *Composites Science and Technology* 49:439–449.
- Luycker, E. D., Benson, D. J., T. Belytschko, Y. B., & Hsu., M. C. 2011. X-fem in isogeometric analysis for linear fracture mechanics. *International Journal for Numerical Methods in Engineering* 87:541–565.
- Nguyen, V. P., Kerfriden, P., & Bordas, S. P. A. 2013. Two-and three-dimensional isogeometric cohesive elements for composite delamination analysis. *Composite Part B* pp. –.
- Piegl, L. & Tiller, W. 1997. The NURBS Book (Monographs in Visual Communication). Springer-Verlag Berlin Heidelberg, 2 edition.
- Turon, A. 2006. Simulation of delamination in composites under quase-static and fatigue loading using cohesive zone models. Engenharia, Universitat de Girona, Spain.
- Verhoosel, C. V., Scott, M. A., Hughes, T. J. R., & de Borst., R. 2011. An isogeometric analysis approach to gradient damage models. *International Journal for Numerical Methods in Engineering* 86:115–134.
- Xu & Needleman 1994. Numerical simulation of fast crack growth in brittle solids. *Journal of Mechanics and Physics of Solids* 42:1397–1434.