Interactive Modeling of NURBS for Isogeometric Analysis

João Carlos L. Peixoto[1], Rafael L. Rangel[2], Luiz F. Martha[1]

[1]*Dept. of Civil and Environmental Engineering, Pontifical Catholic University of Rio de Janeiro (PUC-Rio)*
R. Marquês de São Vicente, 225, 22451-900, Gávea, Rio de Janeiro - RJ, Brazil
joaocpeixoto@tecgraf.puc-rio.br, lfm@tecgraf.puc-rio.br
[2]*International Centre for Numerical Methods in Engineering (CIMNE)*
Carrer del Gran Capità s/n, UPC Campus Nord, 08034 Barcelona, Spain
rrangel@cimne.upc.edu

Abstract. Isogeometric Analysis (IGA) is a numerical analysis method for structures that arises with the proposal of unification between design and simulation, allowing the creation of computational models that preserve the exact geometry of the problem. This approach is possible by a class of mathematical functions called NURBS (Non-Uniform Rational B-Splines), widely used in CAD (Computer Aided Design) systems for modeling curves and surfaces. In isogeometric analysis, the same functions representing the geometry approximate the field variables. In this context, this work was developed to provide a tool within the scope of computational mechanics for two-dimensional isogeometric analysis of linear elasticity problems, including the steps of modeling, analysis, and visualization of results. The system consists of two software programs: FEMEP (Finite Element Method Educational Computer Program - https://github.com/lfmartha/FEMEP), developed in Python and responsible for the geometric modeling stage, and FEMOOLab (Finite Element Method Object Oriented Laboratory - https://github.com/rlrangel/FEMOOLab), a MATLAB software for analysis and display of results. The proposed tool features a graphical user interface (GUI) that allows intuitive visualization and manipulation of NURBS curves with advanced modeling features such as curve intersection and region recognition features that streamline and simplify the process. The open-source system allows collaboration and continuous development by the community of users and developers.

Keywords: Interactive Modeling, Isogeometric Analysis, NURBS.

1 Introduction

An advanced numerical analysis technique known as Isogeometric Analysis (IGA) has recently aroused interest and has been increasingly used in several areas, such as aerospace engineering, mechanical engineering, civil engineering, computational biology, and the automobile industry, among others [1, 2]. This numerical analysis method was initially developed by Hughes et al. [2] and proposed to offer an alternative to the traditional Finite Element Method. Isogeometric analysis offers a different approach, using mathematical functions called NURBS to represent the geometry. The same class of functions is used to approximate the field variables of the analysis. This reduces mesh approximation errors, as the geometry is accurately preserved during analysis.

To assist the academic community, interactive software was developed for two-dimensional geometric modeling of multiple NURBS patches and an additional tool for isogeometric analysis in the context of linear elasticity with plane stress states. The 2D solid modeler is called FEMEP, developed within the same line of research by Bomfim [3], based on the HETOOL library [4], for modeling isoparametric finite element models. The implementation was carried out in Python using the object-oriented programming (OOP) paradigm and has a graphical user interface. The analysis of the models generated by FEMEP is performed by the tool called

FEMOOLab, implemented in MATLAB under the OOP architecture. Similarly, to FEMEP, the FEMOOLab software was initially designed for isoparametric finite element analysis. The incorporation of IGA began with this research, with some publications carried out [5, 6].

The text is organized into six sections. Section 2 introduces concepts of NURBS and B-Splines and section 3 deals with T-Splines. Bezier extraction, an important technique for AIG to determine the basis functions, is presented in Section 4. In sequence, Section 5 discusses the developed software, with some explanations of the architectures and functionalities, and also the presentation of a T-Splines model generated by the developed tool. Lastly, Section 6 encapsulates the article with conclusive remarks.

2 B-Splines and NURBS

B-Spline basis functions are defined from a knot vector, a vector of non-negative parametric coordinates organized in crescent order. A knot vector is represented in the form $\Xi = [\xi_1, \xi_2, \cdots, \xi_{n+p+1}]$, where n is the number of basis functions, p is the polynomial degree, and ξ is the parametric coordinate of the index $i = 1, 2, ..., n + p + 1$. Given a knot vector, a set of B-Splines basis functions is defined for a degree p. For a value of $p = 0$, the basis functions are determined by a constant piecewise equation:

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

For $p > 0$, the B-Splines basis functions are determined by the recursive Cox-de Bor formula:

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{2}$$

Knot vectors can contain repeated values, and when this occurs, the number of times a knot is repeated is called its multiplicity. Open knot vectors are defined as those with initial and final knots having a multiplicity of $p+1$. Open knot vectors are standard in CAD systems and IGA.

A B-Splines curve is defined by a linear combination of a set of functions $\{N_{i,p}(\xi)\}_{i=1}^{n}$, defined from a knot vector $\Xi = [\xi_1, \xi_2, \cdots, \xi_{n+p+1}]$, and a set of control points $\{\boldsymbol{P}_i\}_{i=1}^{n}$:

$$\boldsymbol{C}(\xi) = \sum_{i=1}^{n} N_{i,p}(\xi) \boldsymbol{P}_i \tag{3}$$

From two sets of basis functions $\{N_{i,p}(\xi)\}_{i=1}^{n}$ and $\{M_{j,q}(\eta)\}_{j=1}^{m}$, defined respectively by knot vectors $\Xi^1 = [\xi_1, \xi_2, \cdots, \xi_{n+p+1}]$ and $\Xi^2 = [\eta_1, \eta_2, \cdots, \eta_{m+q+1}]$, and a control net $\{\boldsymbol{P}_{i,j}\}_{i=1}^{n}$ $j = 1, ..., m$, a B-Spline tensor product surface is defined by:

$$\boldsymbol{S}(\xi, \eta) = \sum_{i=1}^{n} \sum_{j=1}^{m} N_{i,p}(\xi) M_{j,q}(\eta) \boldsymbol{P}_{i,j} \tag{4}$$

NURBS entities are an extension of B-Splines, which allow the association of weights at each control point. These weights are used to define rational basis functions. NURBS allow the exact construction of conical shapes, such as circles and ellipses [1]. Univariate and bivariate NURBS basis functions are defined respectively in eq. (5) and eq. (6).

$$R_{i,p}(\xi) = \frac{N_{i,p}(\xi) w_i}{\sum_{\hat{i}=1}^{n} N_{\hat{i},p}(\xi) w_{\hat{i}}} \tag{5}$$

$$R_{i,j}^{p,q}(\xi, \eta) = \frac{N_{i,p}(\xi) M_{j,q}(\eta) w_{i,j}}{\sum_{\hat{i}=1}^{n} \sum_{\hat{j}=1}^{m} N_{\hat{i},p}(\xi) M_{\hat{j},q}(\eta) w_{\hat{i},\hat{j}}} \tag{6}$$

where w_i and $w_{i,j}$ are the weight associated with a control point $\boldsymbol{P}_i$ and $\boldsymbol{P}_{i,j}$, respectively.

2.1 Refinements

A desired feature for geometric modeling and analysis is the ability to refine, especially when seeking more accurate results. In this sense, some algorithms for refining B-Splines and NURBS modify knot vectors and control points, maintaining the original geometry. Piegl and Tiller [7] offer an in-depth analysis of this subject in The NURBS Book.

2.2 Knot Insertion

One of the types of refinement consists of the knot insertion process, which is analogous to the h-refinement in FEM. In a NURBS curve, each element is defined between knots of distinct and consecutive values. Therefore, inserting new knots implies increasing the number of elements. To perform this modification without changing the original geometry of a NURBS, it is necessary to compute a new set of control points.

Given a knot vector $\Xi = [\xi_1, \dots, \xi_k, \xi_{k+1}, \dots, \xi_{n+p+1}]$ the goal is to insert a knot $\bar{\xi} \in [\xi_k, \xi_{k+1})$ to obtain a new knot vector $\bar{\Xi} = [\bar{\xi}_1, \dots, \bar{\xi}_k, \bar{\xi}, \bar{\xi}_{k+2}, \dots, \bar{\xi}_{m+p+1}]$ with $m = n+1$. The basis functions $\{N_{i,p}(\xi)\}_{i=1}^{n}$ are redefined, maintaining the same polynomial order p. A new set of functions $\{\bar{N}_{i,p}(\xi)\}_{i=1}^{m}$ is calculated using eq. (1) and eq. (2) using $\bar{\Xi}$. Then, the new control points are determined $\{\bar{\boldsymbol{P}}_i\}_{i=1}^{n}$, based on the previous control points $\{\boldsymbol{P}_i\}_{i=1}^{n}$:

$$\bar{\boldsymbol{P}}_i = \begin{cases} \boldsymbol{P}_1 & i = 1 \\ \alpha_i \boldsymbol{P}_i + (1-\alpha_i)\boldsymbol{P}_{i-1} & 1 < i < m \\ \boldsymbol{P}_n & i = m \end{cases} \tag{7}$$

where the coefficients α_i are given by:

$$\alpha_i = \begin{cases} 1 & i = k-p \\ \dfrac{\bar{\xi} - \xi_i}{\xi_{i+p} - \xi_i} & k-p+1 \le i \le k \\ 0 & i \ge k+1 \end{cases} \tag{8}$$

2.3 Degree Elevation

Degree elevation refinement is analogous to p-refinement in FEM. The degree elevation formulation is proposed by Piegl and Tiller [8]. This approach can be divided into the following steps: decompose the NURBS curve into Bezier curves, perform the polynomial degree elevation in each Bezier, and remove unnecessary knots.

A NURBS curve can be represented as a chain of Bezier curves, joined end to end. To perform the first step, simply apply knot insertion (Section 2.2) until all internal knots have multiplicity $m = p$, resulting in continuity C^0 at these locations. Subsequently, the control points of each Bezier are extracted. Each Bezier has $p+1$ control points, such that the last point of a given Bezier coincides with the first point of the next adjacent Bezier. The degree elevation is then carried out individually on the Beziers curves. With the degree elevation performed, the new Beziers control points can be regrouped to define the original NURBS, taking care to eliminate repeated control points. The next step is to apply knot removal to this new configuration. The knots initially inserted to obtain the Beziers are removed. The process of removing a knot ξ_k consists of determining a new series of control points. The new knot vector of the NURBS curve includes all the initial knots with their multiplicity increased by one.

2.4 Coons Surfaces

Coons surfaces are a specific type of surface that can be expressed in terms of NURBS. They can be particularly useful as their construction involves defining four boundary NURBS curves. Some restrictions for the contour curves are necessary to make it possible to create a Coons surface [7]. Opposite curves must have the same polynomial degree and the same knot vector.

Let's say we want to construct a Coons surface defined by curves $\boldsymbol{C}_{South}(\xi)$ and $\boldsymbol{C}_{North}(\xi)$, with knot vector $\Xi^1 = [\xi_1, \xi_2, \ldots, \xi_{n+p+1}]$, and curves $\boldsymbol{C}_{West}(\eta)$ and $\boldsymbol{C}_{East}(\eta)$, with knot vector $\Xi^2 = [\eta_1, \eta_2, \ldots, \eta_{m+q+1}]$. The four curves must form a closed region. Therefore, the equation of a Coons surface is given by:

$$\boldsymbol{S}_{Coons}(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} N_{i,p}(\xi) M_{j,q}(\eta) \boldsymbol{P}_{i,j}^{Coons} \tag{9}$$

where $\{N_{i,p}(\xi)\}_{i=1}^{n}$ and $\{M_{j,q}(\eta)\}_{j=1}^{m}$ are the sets of basis functions derived from Ξ^1 and Ξ^2, respectively. The Coons surface control points are given by:

$$\boldsymbol{P}_{i,j}^{Coons} = \boldsymbol{P}_{i,j}^{1} + \boldsymbol{P}_{i,j}^{2} + \boldsymbol{P}_{i,j}^{3} \tag{10}$$

where $\boldsymbol{P}_{i,j}^{1}$ are the control points of a ruled surface, defined by curves $\boldsymbol{C}_{South}(\xi)$ and $\boldsymbol{C}_{North}(\xi)$ in the ξ direction, and by linear curves with knot vector $[0,0,1,1]$ in η direction. Similarly, $\boldsymbol{P}_{i,j}^{2}$ are the control points of a ruled surface, defined by linear curves with knot vector $[0,0,1,1]$ in the ξ direction, and by curves $\boldsymbol{C}_{West}(\eta)$ and $\boldsymbol{C}_{East}(\eta)$ in η direction. $\boldsymbol{P}_{i,j}^{3}$ are the control points of a linear surface, defined by linear curves with knot vector $[0,0,1,1]$ in both directions. The refinements in Sections 2.2 and 2.3 must be performed until all surfaces reach the same knot vectors, ensuring that all three surfaces have the same number of control points. The process of constructing a Coons surface in NURBS form is illustrated in Figure 1.

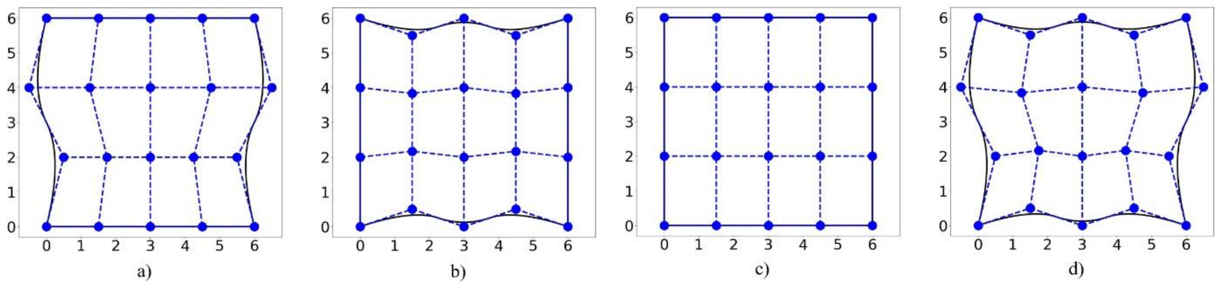


Figure 1. Construction of a Coons surface using NURBS. a) Ruled surface in ξ. b) Ruled surface in η. c) Bilinear surface. d) Coons surface.

3 FEMEP: Software Design

The FEMEP program is essentially based on the HETOOL library [4]. HETOOL is a data structure implemented in Python in the object-oriented programming (OOP) paradigm for iterative two-dimensional geometric modeling. The application is based on the well-known Half-Edge data structure for 2-manifold Solids, adapted for 2D models of planar subdivisions. The library provides the attribute management feature, carried out through a JSON extension file. Bomfim [3] offers a detailed analysis of this topic. In developing the initial version of FEMEP, it was decided to include predefined curves for modeling only straight and polygonal line types. In this work, the tool was expanded to incorporate NURBS-type parametric curves, which are used to generate Coons surfaces and enable two-dimensional isogeometric analysis.

The graphical user interface (GUI) was developed using the Qt framework. In Python, communication with Qt is carried out through the PyQt5 library, which uses the concept of OOP to create and manipulate interface elements (Widgets), known as QtWidgets in Qt. Each QtWidget has a subclass with specific properties and methods. Menus, buttons, check boxes, and scroll boxes are examples of these interface elements. In FEMEP, the visualization of the model and its attributes is performed by OpenGL through the PyOpenGL library. OpenGL is an API for rendering objects, commonly used in fields such as computer graphics and scientific visualization.

FEMEP is a program composed of several modules that allow the user to provide initial information through its interface. The input information can be provided in two ways: by mouse events within the modeling area, managed by the Canvas class, or by triggering interface elements, managed by the AppController. FEMEP is structured around three main classes, following the MVC pattern. The HeModel class stores model data within the scope of geometric and topological entities. HeView accesses HeModel to collect data and provides it to Canvas for on-screen viewing. HeController is the central class, which has access to most of the program's modules and the capability to modify the HeModel. Figure 2 illustrates the robustness diagram formed by the main classes.

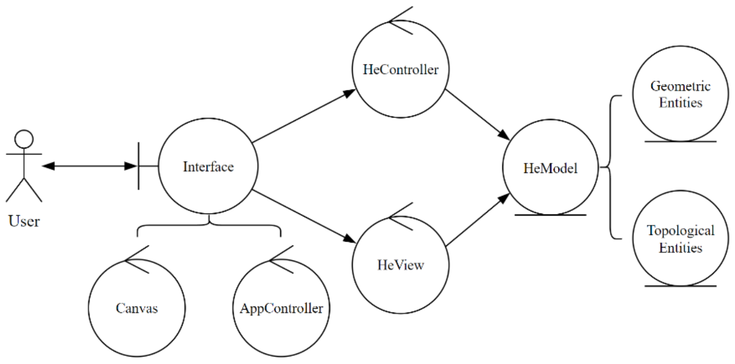


Figure 2. FEMEP robustness diagram.

3.1 Interative modeling of curves

FEMEP has a modeling bar that contains Toggle Button interface elements for modeling predefined curves. When one of these buttons is selected, it enables the modeling of the chosen entity. The curves present in the program are straight line segments, polygonal lines, cubic B-Spline, circle, arc of circle, ellipse, and arc of ellipse. The user can model the curves by clicking the left mouse button in the modeling area, to define points that compose it, called definition points. In addition, a button dedicated to creating points in the model is also present on the modeling bar. Figure 3 illustrates the sequence of definition points for creating each type of curve. Alternatively, the user can input the definition points using the keyboard in the right-hand panel.

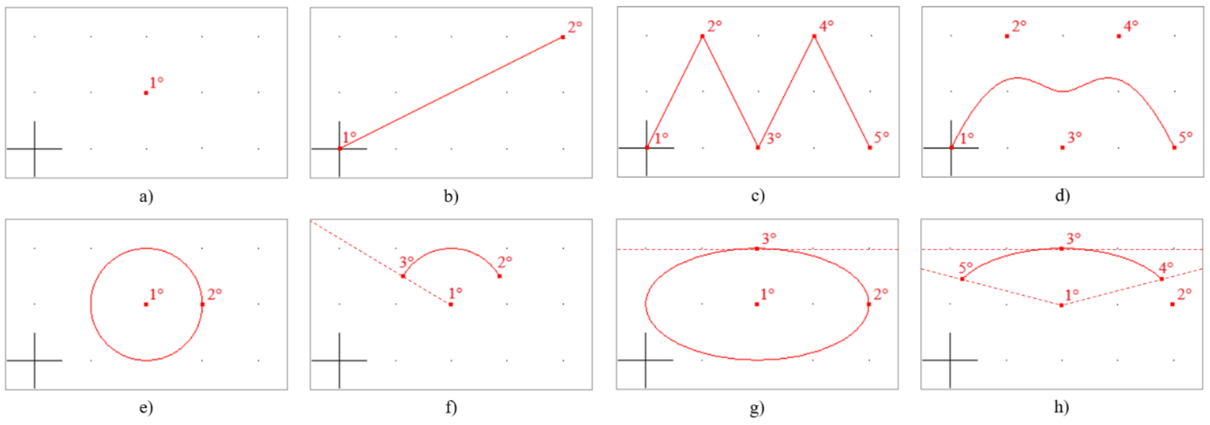
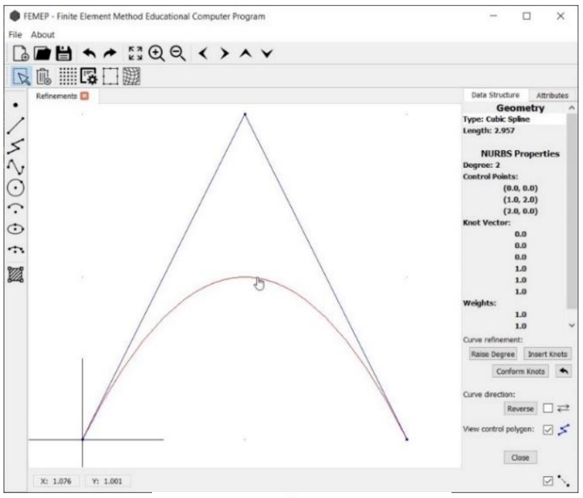


Figure 3. Sequence of definition points required to create each entity.

All curves are treated as NURBS by FEMEP, presenting properties of polynomial degree, knot vector, control points, and associated weights. NURBS information is stored as an object from the Python geomdl library. After selecting a curve, some interface elements are available in the right-hand panel for manipulating and visualizing NURBS properties, as shown in Figure 4a. The “Raise Degree” button is responsible for applying refinement by degree elevation. Similarly, the “Insert Knots” button performs the refinement by inserting knots. The "Conform Knots" button is useful when curves of the same degree need to have the same knot vector. The user should select multiple curves before pressing this button. There are also checkboxes to show an indicator for the curve's direction and to display the control polygon.

Recognition of closed regions is one of FEMEP's modeling features. Simply detecting a closed region does not automatically make it a NURBS surface. Coons surface construction offers an excellent approach in this regard, where the univariate NURBS properties of the contour curves are used to determine the bivariate NURBS properties of the region. The process of generating a Coons surface in FEMEP is carried out using the mesh generation button, and the “isogeometric” option must be chosen. It should be noted that the user must previously guarantee the compatibility conditions of the opposite curves. As a result of generating a Coons surface, the region now possesses NURBS properties that can be visualized in the right-hand panel, as shown in Figure 4b. Additionally, the control points of the Coons NURBS surface can also be displayed.

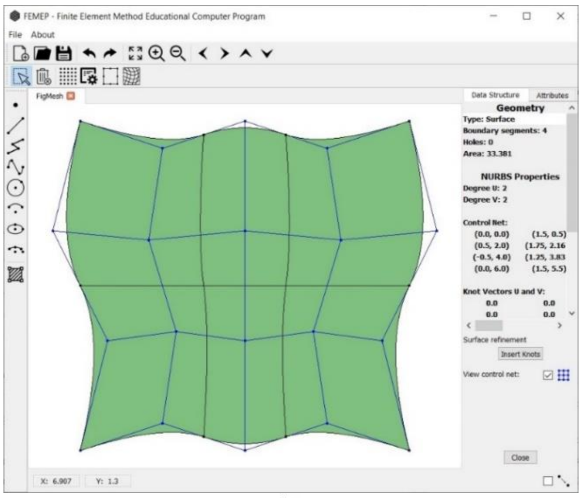


Geometry
Type: Cubic Spline
Length: 2.957

NURBS Properties
Degree: 2
Control Points:
(0.0, 0.0)
(1.0, 2.0)
(2.0, 0.0)
Knot Vector:
0.0
0.0
0.0
1.0
1.0
1.0
Weights:
1.0
1.0

Curve refinement: Raise Degree | Insert Knots | Conform Knots
Curve direction: Reverse ☐
View control polygon: ☑
Close

a)



Geometry
Type: Surface
Boundary segments: 4
Holes: 0
Area: 33.381

NURBS Properties
Degree U: 2
Degree V: 2

Control Net:

(0.0, 0.0)	(1.5, 0.5)
(0.5, 2.0)	(1.75, 2.16
(-0.5, 4.0)	(1.25, 3.83
(0.0, 6.0)	(1.5, 5.5)

Knot Vectors U and V:

0.0	0.0
0.0	0.0

Surface refinement: Insert Knots
View control net: ☑
Close

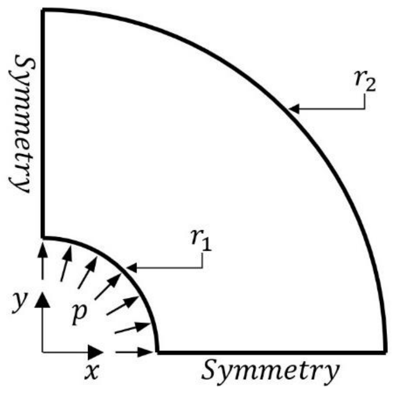
b)

Figure 4. Properties and modeling features. a) NURBS curve. b) NURBS surface.

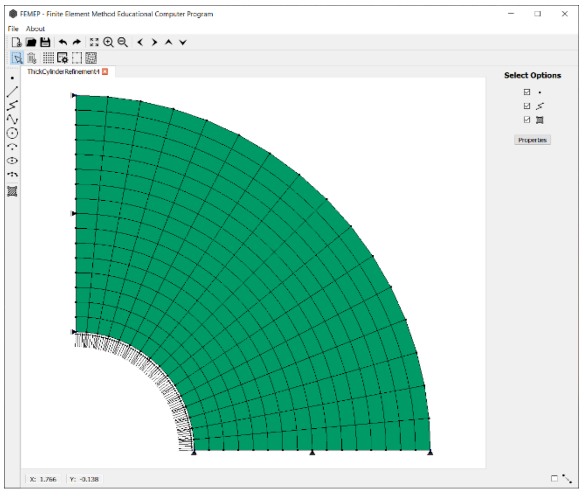
3.2 Numerical Example

The tool is presented with a linear elastic static analysis model, considering a plane stress state. The problem consists of a thick-walled cylinder under constant internal pressure, as illustrated in Figure 5a. Due to symmetry, the modeling is carried out considering only a quarter of the cylinder. The internal radius r_1 is 1 m, and the external radius r_2 is 3 m, with a wall thickness $t = 1$ m. The material properties are specified as an elastic modulus $E = 4.10^{-7}$ kN/m² and a Poisson's ratio $\nu = 0.25$. The applied load is 10 kN/m.

Figure 5b shows the problem modeled in FEMEP using a biquadratic NURBS surface patch. Circle arc boundary curves are defined in the ξ direction, and straight-line boundary curves are defined in the η direction. Degree elevation is employed to achieve quadratic polynomial order.



a)

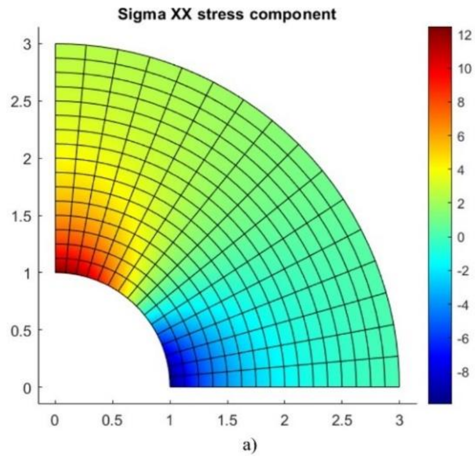


b)

Figure 5. a) Thick-walled cylinder problem. b) Thick-walled cylinder modelled in FEMEP.

With the FEMOOLab program, the σ_{xx} stress distribution was obtained, as presented in Figure 6a. The maximum value for this component occurs at the point $(r = 1, \theta = \pi/2)$. At this location, the analytical solution provides the radial stress component $\sigma_{rr} = -10$ kN/m² and the tangential component $\sigma_{rr} = 12.5$ kN/m² [9]. The stress in the x direction assumes $\sigma_{xx} = 12.5$ kN/m².

The relation between the maximum σ_{xx} stress and the number of degrees of freedom, for more refined models by knot insertion, is given by the graphic in Figure 6b. A Q8 finite element model is also considered. In all scenarios, the results were consistent with the analytical solution. The finite element models exhibited slower convergence, whereas the isogeometric analysis models demonstrated more efficient convergence.



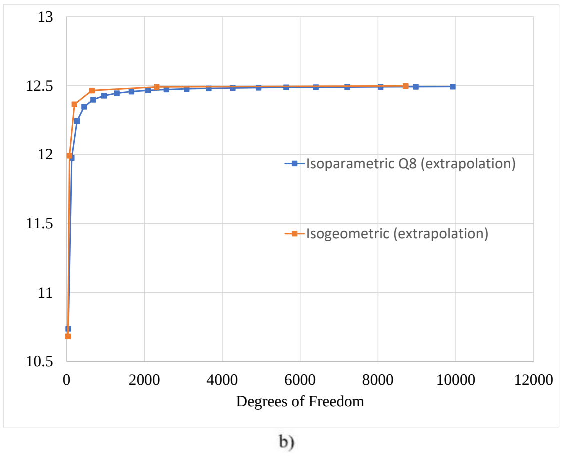


Figure 6. a) σ_{xx} stress distribution on FEMOOLab. b) Graphic of maximum σ_{xx} stress.

4 Conclusions

The modeler stands out for its ability to perform interactive curve modeling, allowing the visualization of NURBS properties and the execution of refinements. The possibility of generating NURBS surfaces from four boundary curves using the Coons concept is a notable feature that enables two-dimensional isogeometric analysis. The programs were verified using a classical problem. The results of stress components obtained from the isogeometric analysis not only converged but also proved superior to a finite element model with Q8 elements.

The OOP architecture resulted in an organized code, which facilitates future expansion. The open-source nature of the tools allows researchers and developers in computational mechanics to contribute to the improvement of the system. For future development, it is interesting to consider implementation in a single integrated tool that encompasses all stages of the isogeometric analysis process.

Acknowledgements. The authors acknowledge the financial support from CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) – Fellowship Productivity in Research of the last author.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors or has the permission of the owner PUC-Rio.

References

[1] COTTRELL, J. Austin; HUGHES, Thomas J.R.; BAZILEVS, Yuri. Isogeometric analysis: toward integration of CAD and FEA. John Wiley & Sons, 2009.
[2] HUGHES, Thomas J.R.; COTTRELL, John A.; BAZILEVS, Yuri. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Computer methods in applied mechanics and engineering, v. 194, n. 39-41, p. 4135-4195, 2005.
[3] BOMFIM, D.S. An Open and Extensible Modeling Strategy for Creating Planar Subdivision Models for Computational Mechanics. Master's Thesis in Civil Engineering - Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, 2022.
[4] BOMFIM, Danilo S. et al. HETOOL: A Half-Edge Topological Object-Oriented Library for generic 2-D geometric modeling. SoftwareX, v. 21, p. 101307, 2023.
[5] PEIXOTO, J. C. L.; RANGEL, R. L.; MARTHA, L. F.; Isogeometric analysis with interactive modeling of multi-patches NURBS. Proceedings of the XLIV Ibero-Latin-American Congress on Computational Methods in Engineering, O Porto, 2023.
[6] PEIXOTO, J. C. L. Isogeometric analysis with interactive modeling of multiple NURBS and T-Splines patches. Master's Thesis in Civil Engineering - Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro, 2024.
[7] PIEGL, Les; TILLER, Wayne. The NURBS book. Springer Science & Business Media, 1996.
[8] PIEGL, Les; TILLER, Wayne. Software-engineering approach to degree elevation of B-spline curves. Computer-Aided Design, v. 26, n. 1, p. 17-28, 1994.
[9] NEGI, L. S.; Strength of Materials. McGraw Hill, New Delhi, 2008.