XLVI CILAMCE
VITÓRIA - ES - BRAZIL - 2025
XLVI IBERO-LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGENEERING

A tool for post-processing of isogeometric laminated structures models

Francisco Rodrigo Pereira da Silva[1], Elias Saraiva Barroso[1]

[1]*Computational Mechanics and Visualization Laboratory, Department of Structural Engineering and Civil Construction, Federal University of Ceará*
Ac. Público, 728 - Pici, 60020-181, Ceará, Fortaleza
rodrigopersi.2016@alu.ufc.br, elias.barroso@ufc.br

Abstract. Composite materials are made by the combination of two or more constituent materials to obtain a strength structure, e.g. reinforced concrete and laminated structures. Isogeometric Analysis (IGA) is a computational approach allowing combining the Finite Element Method (FEM) within a Computer-Aided Design (CAD) applications. The FAST is an academic software to perform numerical simulations via FEM and IGA [1]. This work presents a tool for post processing to visualize the FAST output files, especially for isogeometric laminated plates and shells modeled using rational Bézier elements. The code, written in Python, uses the Matplotlib rendering library and the PyQt6 graphical interface library. The software features are demonstrated through examples of laminated plate and isotropic shell analyzed by FAST, where the distribution of stresses and displacements of the structure is drawn using a contour color map.

Keywords: Composites, Simulation Post-processing, Rational Bézier Elements, Laminated Structure.

1 Introduction

Composite materials are obtained by joining two or more materials on a macroscopic scale in order to produce a material with superior mechanical properties in comparison to its individual components (Barroso [2]). A laminated structure is obtained by arrange a set of composite layers with different orientations, which can result in an optimum use of the material's mechanical, physical and chemical properties. An effective way to analyzing the behavior of these structures is through computer simulations.

The most widely used computational method for analyzing solids and structures is the Finite Element Method (FEM). In its isoparametric formulation, the same functions that describe the displacements of the elements are used to approximate their geometry. Isogeometric Analysis (IGA) is a recent method that applies the isoparametric concept inverted, by approximating the displacements using functions describing by the geometry, which are the same functions used in CAD (Computer-Aided Design) systems (e.g., rational Bézier curves and surfaces and NURBS). This eliminates the error due to the geometry approximation, in contrast to FEM formulations.

FAST (Finite element AnalySis Tool) is a mechanical simulation processing tool developed at Computational Mechanics and Visualization Laboratory (Laboratório de Mecânica Computacional e Visualização) from the Federal University of Ceará. FAST implements both FEM and AIG methods, performing structural analysis of static, dynamic, linear and non-linear problems. The analysis carried out by FAST writes a simulation output file containing the simulation's mechanical responses (e.g., displacements, stresses, and internal forces). There are various post-processing tools for FEM models [3, 4]. However, these tools cannot be used directly for post-processing isogeometric models, which have high-order elements and non-interpolary basis functions.

Therefore, in this work, a post-processing tool was developed, capable of reading and interpreting the FAST output file for isotropic and laminated structural models based on Rational Bézier elements. The software allows visualization of the model used, such as the surface of the elements and the model's control points, as well as heat map of the simulation results, including the structure displacement fields, stresses and internal forces responses. The use of the software is demonstrated through a structural simulation of a laminated plate and an isotropic cylindrical shell.

2 Rational Bézier Surfaces

A *p*th-degree Bézier curve consists of a continuous field of linear combinations of control points $\mathbf{p}_i$ (Barroso [2]):

$$c(\xi) = \sum_{i=1}^{p+1} B_{i,p}(\xi)\, \mathbf{p}_i, \tag{1}$$

where the Bernstein polynomials $B_{i,p}$ are basis functions of degree p, dependent on the parametric coordinate ξ. Regular Bézier curves cannot represent conic curves exactly, e.g., circles and ellipses, which are not described by polynomial functions. Rational Bézier curves consider weights in its formulation and can represent more complex geometries, including conics:

$$c(\xi) = \frac{\sum_{i=1}^{p+1} w_i\, B_{i,p}(\xi)\, \mathbf{p}_i}{\sum_{\hat{i}=1}^{p+1} w_{\hat{i}}\, B_{\hat{i},p}(\xi)}, \tag{2}$$

The weights w_i are associated with each control point $\mathbf{p}_i$ and induce an effect of attraction or repuse between the curve and the corresponding control point.

2.1 Rational Bézier surface of bivariant bases

Bivariant rational basis functions can be written as the linear combination of two univariant Bernstein polynomials, weighted by their respective weights. This process generates a two-dimensional parametric space in the domain (ξ,η) for a degree p in the direction ξ and q in the direction η, defining the rational Bézier surface as:

$$s(\xi,\eta) = \sum_{i=1}^{p+1}\sum_{j=1}^{q+1} R_{i,p}(\xi,\eta)\, \mathbf{p}_{ij}, \tag{3}$$

Where the control points $\mathbf{p}_{ij}$ are associated with the bivariate rational basis functions $R(\xi,\eta)_{ij}$. The bases and the weight function $W(\xi,\eta)$ can be calculated by:

$$R(\xi,\eta) = \frac{w_{ij}\, B_{i,p}(\xi)\, B_{j,q}(\eta)}{W(\xi,\eta)}, \qquad W(\xi,\eta) = \sum_{i=1}^{p+1}\sum_{j=1}^{q+1} w_{ij}\, B_{i,p}(\xi)\, B_{j,q}(\eta), \tag{4}$$

2.2 Nodal results fitting for rational Bézier surfaces

Curves and surfaces fitting consists of determining mathematically, in an approximate or interpolated way, a set of control points that approximate the curve or surface in relation to a set of sample points in Cartesian space. In this work, the procedure is used to evaluate the simulation results located at Cartesian points, such as the nodal stresses located at corresponding parametric locations of the Bézier surface element. In this case, the sample points are the nodal stresses obtained by extrapolating the values calculated at the integration points or evaluated in the parametric space of the element. The adjustment process seeks to find the coordinates of the control points that best reproduce these sample points. This procedure can be formulated as a linear model. To do this, each measured point is associated with a corresponding parametric position in the domain of the curve or surface.

Therefore, for a set of n_{sp} sample points, associated with parametric positions $[(\xi_1,\eta_1,\ldots,\xi_{n_{sp}},\eta_{n_{sp}})]$, the equation system is obtained:

$$\begin{bmatrix} \mathbf{b}(\xi_1, \eta_1) \\ \vdots \\ \mathbf{b}(\xi_{n_{sp}}, \eta_{n_{sp}}) \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_{(p+1\times q+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{s}_1 \\ \vdots \\ \mathbf{s}_{n_{sp}} \end{bmatrix} \Rightarrow \mathbf{B}\,\mathbf{p} = \mathbf{s} \tag{5}$$

Where $\mathbf{b}_i$ is a vector of surface basis functions evaluated at (ξ_i, η_i), $\mathbf{p}$ is the fitted results at control points and $\mathbf{s}$ are the results at sample points.

The solution of this system results in a matrix containing the control points corresponding to the stresses of the representation. With results evaluated at control points appropriately, the result field can be approximated using the same surface basis functions defined by Eq. 4.

3 PyNeuVis

The PyNeuVis software was developed as a post-processing tool for analyses done using FAST. The programming language used was Python version 3.9. The following libraries were used: PyQt6, Matplotlib, NumPy, and the Bezier (proposed in this work). Some routines to represent curves and Bézier surfaces were also implemented.

The source code has been structured into four general modules. The *model* module is responsible for reading, interpreting and storing the model data extracted from the results file. The *geom* module consists of methods that define the geometry of the model based on the formulation of rational Bézier curves and surfaces. The *gui* module implements the methods for creating the interactive user interface, and *vis* contains the routines for rendering the models data and its results.

3.1 Rendering library

Matplotlib is a library for creating static, animated and interactive visualizations (Matplotlib [5]). One of the object-oriented tools for 3D plotting in the library is *Poly3DCollection*. The class internally implements a collection of polygons by explicitly storing the coordinates of the vertices of each face. Unlike other functions, e.g. *plot_surface* and *plot_trisurf*, *Poly3DCollection* allows arbitrary geometries. One of the advantages of this approach is that it avoids unnecessary discretizations, which maintains the fidelity of the represented geometry. In addition, its structure avoids conversions and intermediate meshes, reducing processing overhead and memory usage. Fig. 1 shows an example of a triangulated tessellation in PyNeuVis that can be defined by the user using *Poly3DCollection*. This feature makes it possible to adjust the subdivision of the patches and ensures high resolution for the results contour plots, specially for high-order elements.

Another important Matplotlib function used in *VIS* module is *contourf*. This routine creates 2D contour maps from scalar field such as internal forces and displacements. This routine is processed in the parametric space of the element, then the output strips polygons are mapped to the Cartesian space using the surface equation. It is worth pointing out that, without this procedure, it was not possible, in the current implementation of Matplotlib, to generate heat maps for arbitrary scalar fields over surfaces in three-dimensional space.

3.2 Graphical Interface

The *Qt toolkit* is a comprehensive set of libraries and tools, originally developed in C++, aimed at creating high-performance graphical interfaces. PyQt6 is the official binding that exposes Qt's functionalities to the Python language, inheriting its native functionalities and widgets (Chandrakar and Bahadure [6]). PyQt6 allows the Matplotlib rendering window to be incorporated into the software interface, which gives the graphics widget an interactive real-time update.

Qt Designer is a tool in the *Qt toolkit* for creating user interfaces visually and intuitively, without having to write code directly. The tool allows the developer to build layouts, buttons, windows and other widgets using the drag-and-drop concept. At the end, a *.ui* file translated into Python is returned, containing the generated GUI. Note that, in the interface in Fig. 1, the checkboxes marked allow the user to view the surface, triangulation, control net and the elements edges.

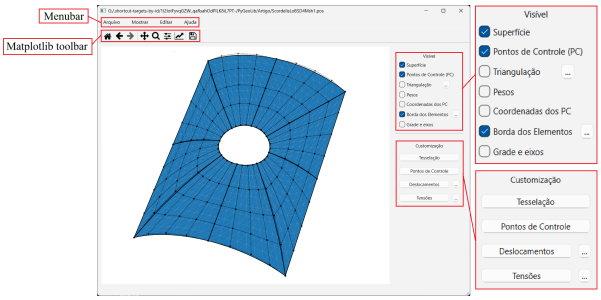


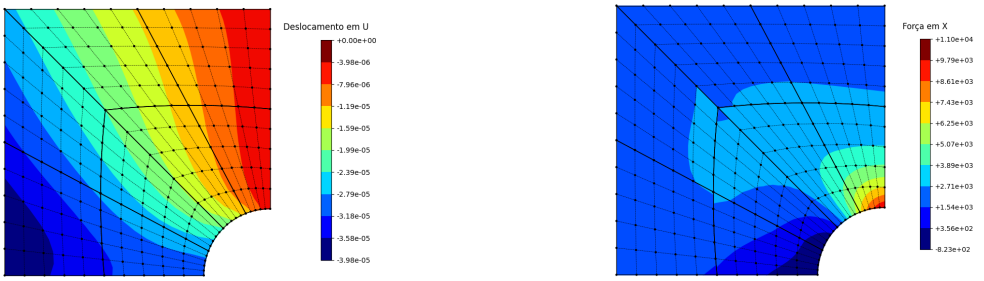
Figure 1. PyNeuVis graphical interface.

4 Application Examples

4.1 Laminated plate with a hole submeted to a traction

The laminated plate analyzed has a symmetric layup with angles $\mathbf{\Theta}$ = [+45° 90 +45°] and thickness $\mathbf{t}$ = [3mm 6mm 3mm]. The material properties of each plate are orthotropic. The modeling considered only a quarter of the plate due to symmetry of the problem. The applied load is a uniform traction of 2500 N/m.

The initial model was carried out using NURBS, as presented in Oliveira and Barroso [7], using 128 quadratic elements. In this work, the structure is analyzed using shallowshell elements based on rational Bézier surfaces, which are based on the formulation presented in Praciano et al. [8]. The displacement field (u) is shown in Fig. 2 a), and internal force in x direction is shown in Fig. 2 b). It can be seen that the level of tessellation (10 divisions in each parametric direction) allows to represent curvilinear strips within the same high-order element. Moreover, it can be noted a force concentration at the hole, with a concentration factor of 4.4, which is an expected behavior for the kind of problem.



(a) Displacement u result. (b) Force in x result.

Figure 2. Visualization in NeuVis of the displacement and internal forces fields for the laminated plate.

4.2 Scordelis-Lo shell with hole

This example consists in the analysis of the Scordelis-Lo shell with a central cutout. The model has 8 quartic general shell rational Bézier elements. The geometry, loading and boundary conditions followed the configuration presented in Barroso [2]. Figure 3 shows the w-displacement feild. The visualization was carried out using the PyNeuVis software.

The maximum displacement w is similar to the reference solution reported in Barroso [2], and the depicted field is similar.

4.3 Conclusions

In this article, post-processing software was developed to visualize FAST results for isogeometric models based on rational Bézier elements. Implemented in Python 3.9, PyNeuVis integrates PyQt6 and Matplotlib to provide an interactive interface and 3D visualization. The application in two test cases validated the tool's capability to represent displacement and stress fields with high resolution. As future work, it is planned to extend support to

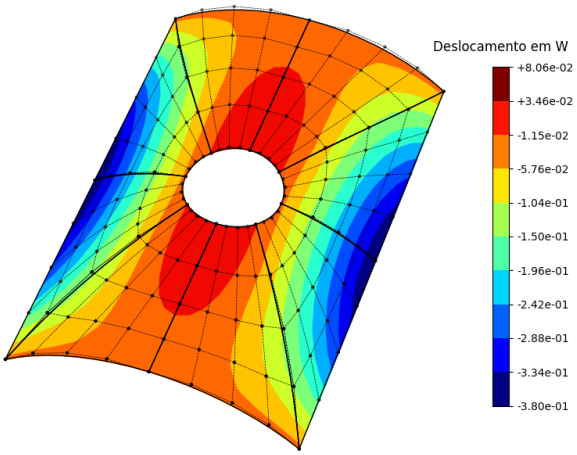


Figure 3. Displacement field w of the Scordelis-Lo shell with hole.

other rational elements, incorporate advanced visualization features and integrate new output formats.

Acknowledgements. The authors acknowledge the financial support from UFC (Universidade Federal do Ceará) - PIBIC 2024/2025 - Announcement Nº 01/2024.

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 owners to be included here.

References

[1] FAST, web page. 2025. *https://pec.ufc.br/index.php/pt-br/downloads*
[2] E. S. Barroso. *Geração de malhas de alta ordem para análise isogeométrica utilizando elementos de Bézier racionais*. PhD thesis, Federal University of Ceará.
[3] ParaView, web page. 2025. *https://www.paraview.org/*
[4] Gmsh, web page. 2025. *https://gmsh.info/*
[5] Matplotlib, web page. 2025. *https://www.matplotlib.org/*
[6] Chandrakar, S.; Bahadure, Dr. N. B. *Python GUI with PyQt: Learn to build modern and stunning GUIs in Python with PyQt5 and Qt Designer*. 1st. ed. 2024.
[7] Oliveira, M. A.; Barroso, E. S. *Pre-processing for isogeometric analysis of laminated composities*. XLV Ibero-Latin American Congress on Computational Methods in Engineering. Maceió, 2024.
[8] Praciano, J. S. C. et al. *An isogeometric formulation for stability analysis of laminated plates and shallow shells*. Thin-Walled Structures. Volume 143, 2019.