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

Graphical-Interactive Software to Support the Teaching of the Finite Element Method

Mariana Z. Reis[1], Marcos A. C. Rodrigues[1]

[1]*Dept. of Civil Engineering, University of Espírito Santo (UFES)*
Avenida Fernando Ferrari, 514, 29075-910, Vitória/ES, Brazil
mariana.z.reis@edu.ufes.br, marcos.a.rodrigues@ufes.br

Abstract. The Finite Element Method (FEM) is an approximate numerical method widely used in structural analysis. It consists of dividing a complex structure domain into several subdomains with simpler geometry, to solve the entire problem through the solution of its subparts. The association of the method with computational tools is already present in several Engineering programs, which provide results in the form of internal force diagrams, nodal displacements and rotations, and stresses. In Civil Engineering courses, the method is primarily applied to model linear structural elements, also known as matrix structural analysis. However, the correct use of these tools requires previous knowledge of the fundamental theory behind the method, which is not the focus of current available computer software. Therefore, the present study aimed to develop a computational tool designed to provide intermediate steps and the logic of the method in a step-by-step approach, including an interactive graphical interface for structure modeling and visualization of results. The tool was validated with several examples from literature demonstrating its potential for educational purposes.

Keywords: Finite Element Method, Graphical-Interactive Software, Object-Oriented Programming, Educational Software, Python Program.

1 Introduction

The Structural Analysis, according to Martha [1], aims to predict the behavior of a structure to ensure the fulfillment of safety and serviceability requirements. For this purpose, it is fundamental to represent the structure as an idealized model, involving simplifying assumptions regarding its geometry, loads, boundary conditions, and material properties. Once this analytical model is established, numerical methods become indispensable, especially when supported by computer tools capable of handling large systems of equations and complex geometries [2]. In the context, the Finite Element Method (FEM) is widely used in Civil Engineering to solve typical structures of the field. The main idea is to divide the continuous domain of the structure into finite subdomains with simpler geometry, making it possible to solve the entire problem through the solution of its subparts. Although this method is already implemented in software such as ANSYS Mechanical APDL [3], and in free tools like Ftool [4], LESM [5], and MASTAN2 [6], the focus of these and other programs is mainly the achievement of numerical results of internal forces and displacements, diagrams, as well as didactical images with its effects on the structure.

As a matter of fact, these tools simplify daily routines for professionals as they solve long and difficult equations quickly, allowing them to focus on the engineering of the problem. However, intelligent software use requires understanding the theoretical basis of the method involved to ensure correct application or adaptation [2]. From this perspective, this work aims to develop a computational tool – StifFEM – with an educational purpose and an easy-to-understand design, intended to demonstrate the intermediate steps of a Finite Element analysis and to contribute both to classroom learning and to engineering designs. Built using Object-Oriented Programming (OOP) paradigm for structural analysis [7], using Python [8] language and currently under improvement, the tool includes the following features: a graphical interactive interface, the matrix analysis conducted by the method - usually hidden in the aforementioned programs -, as well internal forces, displacements and reactions.

2 A summary of the method

The Finite Element Method (FEM) is used in different areas of Engineering, such as Electromagnetism, Thermodynamics, Fluid Mechanics, and Structural Analysis. The application is characterized by a few steps, including: the discretization of the continuous domain into discrete finite elements, creating a "mesh" (the smaller the elements, the higher the accuracy and the computational cost of the solution); next, the definition of shape functions that approximate each element solution at any point based on the known values at its nodes; then, a known physical principle (such as the Principal of Virtual Work) is applied to each element, allowing the formulation of the continuous problem through the assembly of the shape functions; finally, the global matrix-based system is assembled, considering the boundary conditions and external loads; the system is then solved.

The method approach usually leads into a matrix-based analysis, involving several large matrices in highly discretized meshes, which makes it suitable for computational implementation. To introduce the FEM, a simplified variant commonly used is the Direct Stiffness Method, a matrix formalization of the Displacement Method. According to Martha [1], in the Displacement Method, the primary unknowns are the displacements and rotations at the ends of each element – nodes – while all other variables are expressed in terms of the primary ones. The Direct Stiffness Method precisely assigns a stiffness matrix to each element of the mesh that relates these primary unknowns to the forces or moments acting in each node. Following the FEM methodology, each element matrix is assembled into a global stiffness matrix $[K]$ representing the entire structure, which is then incorporated into a system with the displacement and rotation $\{D\}$ and the generalized force $\{F\}$ vectors, as represented in eq. (1).

$$[K] \cdot \{D\} = \{F\}. \tag{1}$$

Equation (1) is a condensed and representative version of the equilibrium system, as a complete procedure must be considered for obtaining the local matrices of each element, transforming them into the global coordinate system, and formulating the generalized force vectors at each node, until it is finally possible to apply the equilibrium equations and solve the structure. In this context, the computational tool StifFEM aims to assist the construction of this system by providing access to these intermediate steps involved in the analysis.

3 A walk through StifFEM

3.1 Graphical interface

Initially, Fig. 1 presents both the interactive modeling and results interfaces of the program, which were built using the PyQt5 [9] library of the Python programming language.

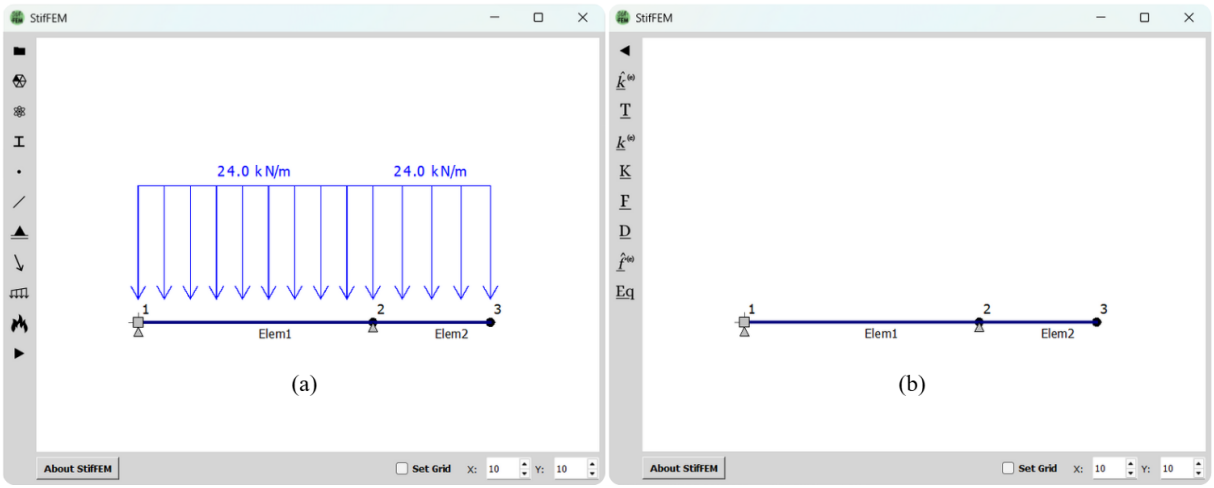


Figure 1. Graphical Interactive Modeling (a) and Results (b) Interface for structural analysis

Both interfaces are part of a main class that triggers a few other secondary classes through the buttons located

in the left sidebar. Each secondary class of the first interface – Fig.1 (a) – is responsible for a specific step in the structural modeling process, opening a small window – known in the PyQt5 as a "dialog" – in which the data related to that step is entered. When the analysis is executed, the second interface – Fig. 1(b) – is loaded with new secondary classes to display intermediate steps and results also through dialogs. According to the order of appearance of the buttons, Tab. 1 and Tab. 2 below provides a brief explanation of their functionality.

Table 1. Overview of modeling interface buttons and associated classes

Button	Class	Function
■	File	Imports a text file containing the structure data for analysis
⬡	Theory and Model	Selection of the beam theory and definition of the structure model
⚛	Material	Young's modulus, Poisson and Thermal expansion definitions
I	Section	Cross-section Area, Moment of Inertia, Dimensions, Form factor
•	Node	Insertion of the node coordinates
/	Element	Connection of the nodes by inserting elements
▲	Support	Assignment of support and prescribed displaceability to nodes
↘	Nodal Load	Assignment of point loads to nodes
⇊	Distributed Load	Assignment of distributed loads to elements
🔥	Temperature	Assignment of temperatures to elements
▶	Run Analysis	Creates a text file with structural data and executes the analysis

Table 2. Overview of result interface buttons and associated classes

Button	Class	Function
◀	Back	Returns to modeling the structure
$\underline{\hat{k}}^{(e)}$	Element Local Matrix	Exhibition of the local stiffness matrix of each element
$\underline{T}$	Element Transformation Matrix	Exhibition of the rotation/transformation matrix of each element
$\underline{k}^{(e)}$	Element Global Matrix	Exhibition of the global stiffness matrix of each element
$\underline{K}$	System Global Matrix	Exhibition of the global stiffness matrix of the structure
$\underline{F}$	System Forces	Exhibition of a vector with all the forces of the structure
$\underline{D}$	System Displacements	Exhibition of a vector with all the displaceability of the structure
$\underline{\hat{f}}^{(e)}$	Element Internal Forces	Exhibition of a vector with the internal forces of each element
$\underline{Eq}$	System Equations	Exhibition of the partitioned (reduced) system

3.2 Programming logic

The first project phase promoted the development of a Python code that implements the Finite Element Method for a structural analysis. To achieve this goal, the program was composed of a central script denominated "main" and several other secondary ones that comported different classes, which were always called within the central code to execute specific FEM related procedures. The second phase of the project, under improvement, involved the creation of the graphical interface using PyQt5, and its integration with the calculus routine, allowing the user to visualize the structure, the intermediate steps (matrix analysis) and the results. The constructed matrices and vectors were then returned to the interface, which subsequently stored and displayed them on screen as requested by the user. The classes diagram is presented according to the UML standard in Fig. 2, on next page.

The "Dialogs" box represents the set of classes responsible for acquiring structural data through the interface, as well as for displaying the resulting variables. During the structure modeling, each button in the "MainWindow" triggers one of these dialogs where the corresponding values are entered. Once provided with the input, the dialog returns these data already semi-processed to the interface, which either displays it in the form of a drawing (nodes, elements, supports) or uses it to perform an analysis. In this scenario, the "Run analysis" button is designated to generate a summary text file and to trigger the "main" structure, which executes the method. Within the "main", the "Read" class reads the file and stores the data into initially null predefined variables. Subsequently, each node and each element are defined as objects using the "Node" and "Element" classes.

Once the FEM procedure was effectively initiated, these created objects served as parameters for the construction of the Local and Global Stiffness Matrices of each element through the "Truss", "Beam", and "Frame" classes – each governed by the specific formulation corresponding to the structural type it represents (truss, beam,

or frame), and to the theory adopted (Euler-Bernoulli or Timoshenko). Immediately after, the "GlobalMatrix" and "ReducedMatrix" classes, assisted by the "Assembler" class, are responsible for building the Global and Reduced Stiffness Matrices of the system, respectively. The following class, "EquivalentNodalForces", converts the temperature gradients and distributes loads applied to the elements into equivalent nodal forces. Subsequently, the "Solver" class handles the solution of the system formed by the Reduced Matrices – which are those implemented in computational simulations – and the known displacement and force vectors, yielding the nodal displacements and support reactions of the global structure. To conclude the method's implementation, the "ElementForces" class uses the previously obtained results to compute the internal forces for each element. All these newly calculated values are then returned to the "MainWindow" and displayed on screen as each results-related button is triggered. Finally, the "Export" class exports a text file containing the initial problem data, the intermediate steps of the method, and the results for internal forces and displacements.

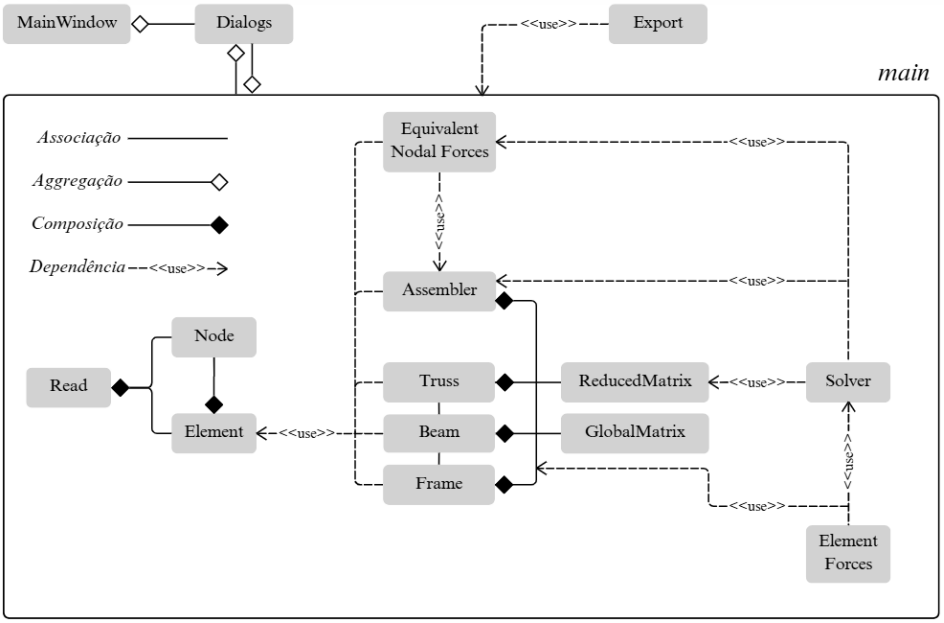


Figure 2. Classes diagram in the UML standard

3.3 Validation

The StifFEM tool was validated mainly through comparison with problems solved in textbooks and manuals covering the Finite Element Method, as well as with other software. One of these examples involved a beam taken from Pitangueira [10] already modeled in the tool as shown in Fig. 1, with an Elasticity Modulus of 200 GPa and a cross-sectional moment of inertia of 0.000125 m⁴. By implementing a direct comparison between the results, the Global Stiffness Matrix of the structure is presented in Fig. 3, support reactions and displacements at each node are shown in Fig. 4, and the two elements internal forces ($p^{(1)}$ and $p^{(2)}$) are presented in Fig. 5.

$$\underline{K} = 2 \times 10^2 \begin{bmatrix} 12 & 30 & -12 & 30 & 0 & 0 \\ 30 & 100 & -30 & 50 & 0 & 0 \\ -12 & -30 & 108 & 90 & -96 & 120 \\ 30 & 50 & 90 & 300 & -120 & 100 \\ 0 & 0 & -96 & -120 & 96 & -120 \\ 0 & 0 & 120 & 100 & -120 & 200 \end{bmatrix}$$

(a)

Global Matrix

You can see the system global stiffness matrix below

	d1y	θ1	d2y	θ2	d3y	θ3
d1y	2400.00	6000.00	-2400.00	6000.00	0.00	0.00
θ1	6000.00	20000.00	-6000.00	10000.00	0.00	0.00
d2y	-2400.00	-6000.00	21600.00	18000.00	-19200.00	24000.00
θ2	6000.00	10000.00	18000.00	60000.00	-24000.00	20000.00
d3y	0.00	0.00	-19200.00	-24000.00	19200.00	-24000.00
θ3	0.00	0.00	24000.00	20000.00	-24000.00	40000.00

(b)

Figure 3. Global Stiffness Matrices by Pitangueira [10] (a) and by StifFEM (b)

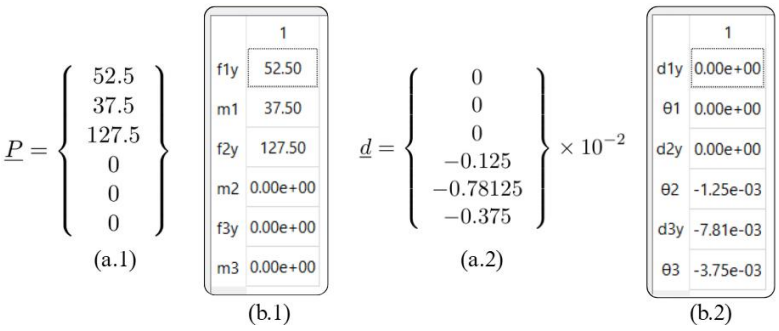


Figure 4. Support reactions and nodal displacements by Pitangueira [10] (a.1 - a.2) and by StifFEM (b.1 - b.2)

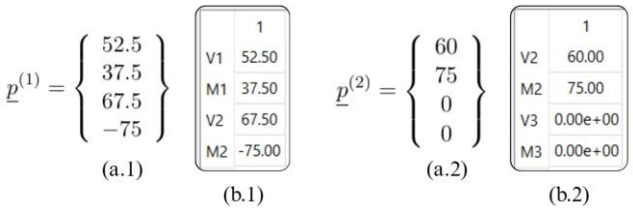


Figure 5. Internal forces by Pitangueira [10] (a.1 - a.2) and by StifFEM (b.1 - b.2) for each element

4 Conclusions

It is concluded that the StifFEM is well-suited for its intended purpose and may serve as a didactic tool for undergraduate and graduate-level teaching of the Finite Element Method. In addition to computing the results like currently available software, showing intermediary steps is a key feature that supports students both during classroom instruction and in independent study. The numerical application evidenced the capability of the software to accurately compute the structural response at validations. Although under improvement, future versions are expected to include features for representing reactions, displacements, and internal forces acting on each element.

Acknowledgements. The authors acknowledge and thank the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) for funding this study.

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] L. F. Martha, *Análise de Estruturas: Conceitos e Métodos Básicos*. GEN LTC, 2022.
[2] S. C. Chapra and R. P. Canale, *Métodos numéricos para engenharia*. AMGH Editor, 2016.
[3] ANSYS Inc., *ANSYS Mechanical APDL*, version 2023. Available at: https://www.ansys.com. Accessed on: Jul. 15, 2025.
[4] L. F. Martha, *Ftool: A graphical-interactive program for teaching structural behavior*. PUC-Rio, Rio de Janeiro, 2000. Available at: https://www.maxwell.vrac.puc-rio.br/124/124_PRINCIPAL.HTM. Accessed on: Jul. 15, 2025.
[5] L. F. Martha, R. L. Rangel, P. C. Lopes, C. Horta, *LESM - Linear Elements Structure Model*. A graphical-interactive program for structural analysis of trusses, frames and grillages. Available at: https://web.tecgraf.puc-rio.br/lesm/index.html. Accessed on: Aug. 4, 2025.
[6] W. McGuire, R. H. Gallagher, R. D. Ziemian, *Matrix structural analysis*. John Wiley & Sons Inc, NY, USA, 2000
[7] L. F. Martha, *Análise matricial de estruturas com orientação a objetos*. Elsevier, Rio de Janeiro, 2018.
[8] Python Software Foundation, *Python Language Site: Documentation*, 2024. Documentation page. Available at: https://www.python.org/doc/. Accessed on: Jul. 18, 2025.
[9] Riverbank Computing, *PyQt5 Reference Guide*. Available at: https://www.riverbankcomputing.com/static/Docs/PyQt5/. Accessed on: Jul. 18, 2025.
[10] R. L. Pitangueira, *Introdução ao Método dos Elementos Finitos*. Version 1.01, 2019. Lecture notes.