XL CILAMCE
IBERO-LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGINEERING
NOVEMBER 11-14, 2019
Praiamar Natal Hotel & Convention
Natal, RN-BRAZIL

BEMGUI: Graphical Interface for Modeling via Boundary Element Method

Tito A. Grossi[1]
Gilberto Gomes[2]

1 tito.brossi@gmail.com; Civil Engineering Student, University of Brasília – Brasília, Brasil
2 ggomes2007@gmail.com; Post-Graduate Program in Structures and Construction – PECC/University of Brasília – Brasília, Brasil;

Abstract. Developing and adapting computation methods for engineering problems, especially the Boundary Element Method (BEM), has been an important area of research. In this context, BEMGUI emerges with the goal of being a graphical user interface (GUI) for the generation of drawings, physical-geometric modeling and contour element meshes, in order to interact with programs that use the BEM to analyze elastostatic problems. It also aims to give a form of visual representation of the results, being a pre and post processing tool. For this, the program is implemented in the Python language and follows the object-oriented programming (OOP) paradigm, as Python adapts well to the application programming interface (IPA) of a GUI. In addition, the flexibility and powerful ideas of this paradigm offers great advantages to the scripting of a program. The result is a fast, interactive GUI that talks to the user through mouse click and mouse movement events, in addition to keyboard keys. User-supplied information is, at the end of a modeling process, provided to programs such as BEMCRACKER2D, which uses BEM to generate results in crack problems, for example.

Keywords: Graphical User Interface; Modeling; Boundary Element Mesh; Python

1. INTRODUCTION

A modelling process is nothing more than a simplified imagination of a real-world object that is usually created to better understand the modeled object [1]. On the other hand, computational modelling has been implemented to solve problems that are too complex or extensive for the human mind [2] and provides advances in a lot of different scientific domains.

Therefore, developing and adapting graphical computing techniques, geometric modelling and data structures to create object-oriented programs (OOP) [3] that allows computing simulation through numeric methods, specially the boundary element method (BEM) has been an important research matter in the civil engineering area.

In this context, BEMGUI is the result of an effort to develop an interactive and user-friendly graphical user interface (GUI) in a free and open-source language and substantiated in OOP for modeling simple and complex elements through the use of mouse and keyboard events. It interacts with BEM numerical programs, like BEMCRACKER2D [4] to provide results and tries to use those to produce a visual representation of the expected behavior.

2. MATERIALS AND METHODS

The development of this project uses the Python language due to its compatible features with the creation of a graphical interface: it is a high-level, general-purpose, easy-to-learn, dynamic typed and interpreted language [5]. Besides that, this language not only supports, but is developed in the OOP paradigm, which is a conjecture of ideas that makes the programming process less repetitive, more readable and better-suited for large scale programs [6]. The version of Python used in this project is 3.5.2 [7].

BEMGUI also has a close relationship with BEMCRACKER2D, a robust academic program written in C++ to analyze two dimensional elastostatic problems using BEM and has 3 types of study: standard BEM, with no crack growth and with crack growth.

3. THE BEMGUI INTERFACE

As stated, the project uses the Python language and the libraries that comes with it. The main tool provided by Python to this work is the external library PyQt in version 5.12.1 [8] and its API. The program aims to be a pre-processer and post-processer software tool, but, in this article, we will restrain ourselves on the pre-processer part of the program.

The BEMGUI interface is presented in the 5 modules listed below and divides those in a simple and clean design model that can be seen in figure 1a. It contains tips for all its functions (figure 1b).

- Geometry: contains three buttons for the creation of geometrical elements, those being point elements (which every other element in this module is dependent of), straight elements, curved elements (arcs, quadradic and cubic curves) and a button to define zones.
- MESH: module used to generate the discretization of straight and curved elements, that will define, at the end, the mesh formed in the construction of the model. There are three methods of meshing (BEM, Finite Element Method, or FEM, and Meshless), but, in this project, we will focus on BEM. The user may discretize an element as a non-crack element (linear progression between points) or as crack elements (geometrical progression between points).
- BOUNDARY CONDITIONS: Applies the surface conditions of the object, that is, displacement restrictions, tractions and unknown conditions.

- ELASTOSTATIC ANALYSIS: Generates 3 types (compatible with the 3 methods used in BEMCRACKER2D) of archives to communicate with BEM programs, that generates numerical results when executed.
- GRAPHICAL RESULTS: Uses the results provided by BEMCRACKER2D to create graphical visualization of those.

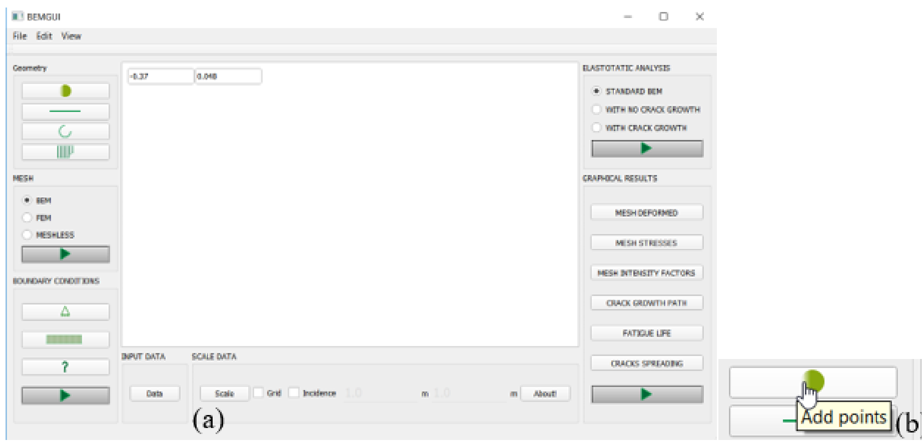

Figure 1: BEMGUI interface (a) the design of BEMGUI; (b) tip presented when mouse is over button

As part of a support to the program, secondary windows were built to help functions that needs more than mouse clicks and movements. These functions are, for example, choosing the sort of curved element the user wants to draw. All the window used are shown in figure 2.

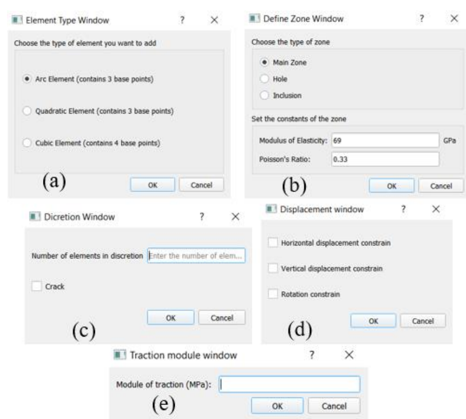

Figure 2: Secondary window used as support to the modeling process: (a) choosing type of curved element to be drawn; (b) defining characteristics of chosen zone; (c) informing the number of points in discretion of element; (d) adding displacement restrictions to elements; (e) defining the module of a traction

BEMGUI is a mouse-event based window and certain functions can only occur when the mouse is over the graphical representation of an element. Thus, it provides help by highlighting those when they can receive mouse events bases on their positions, as figure 3 shows.

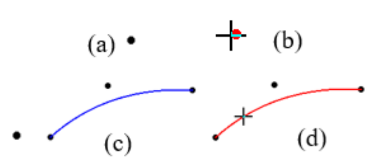

Figure 3: Example of highlight events: (a) point element in normal colors; (b) highlighted point element; (c) curved element in normal colors; (d) highlighted curved element

4. APPLYING BEMGUI

This section will show the construction of a model based on the cracked cruciform plate problem proposed by [9]. Firstly, using the geometry module, we add the base points of the geometrical drawing (figure 4a). With those added to the drawing area, it's possible to create the full geometry, as displayed on figure 4b. From that we can determine the zone and its mechanical characteristics through mouse clicks over elements and the dialog box described in figure 2b.

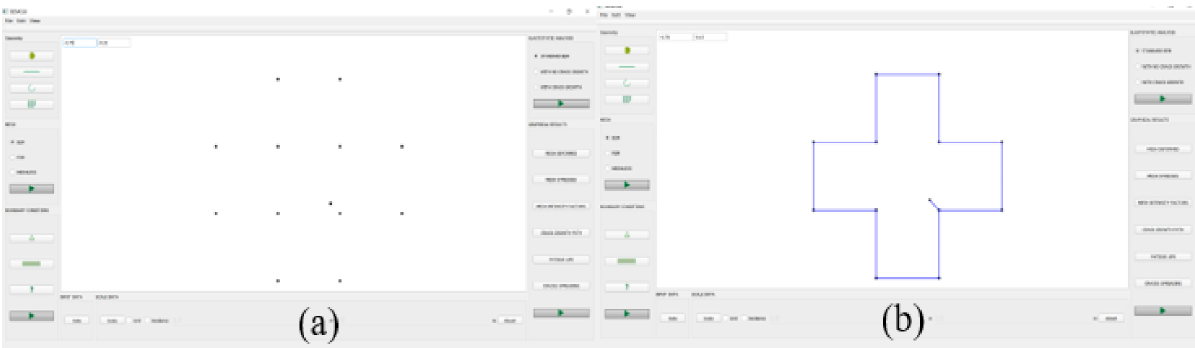

Figure 4: Geometric model of the problem (a) base points; (b) Completed drawing

With the geometrical proportions defined, we can discretize every element though the use of the dialog window in figure 2c. Here, all items were defined as having 9 points of discretization, and, for the crack element, it follows a geometric proportion. The generated mesh is shown in figure 5.

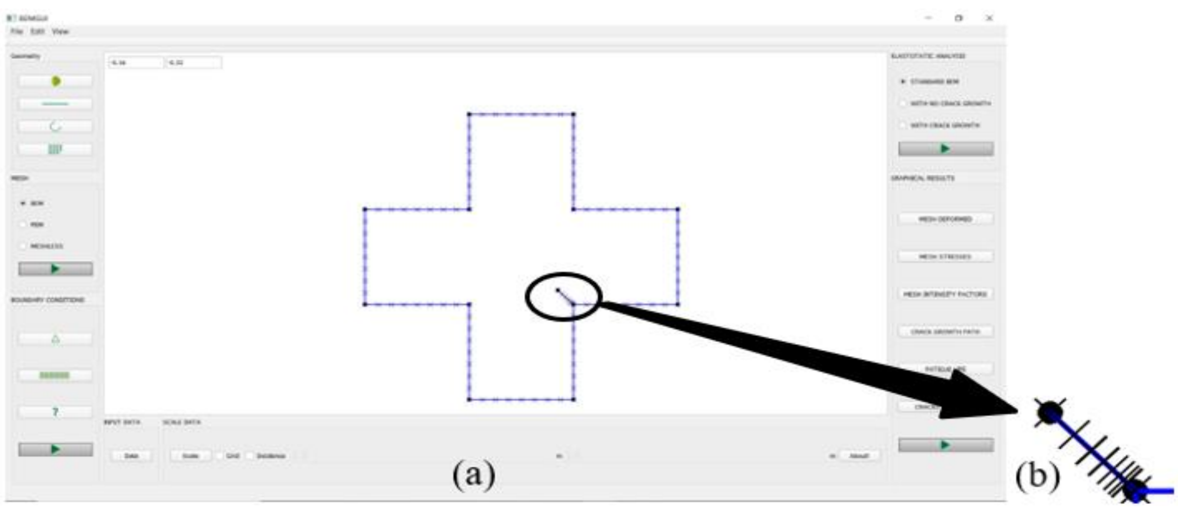

Figure 5: Mesh generation (a) whole mesh generated; (b) detailing of crack, which follows a geometrical proportion in its spacing between mesh discretion

Finally, it's possible to add the object's boundary conditions, by determining traction values using the window shown in figure 2e and movement restrictions with the support of window disposed in figure 2d. With that, the modeling process is ended and the result can be seen if figure 6.

From the constructed model, BEMGUI can then create an archive that informs the case evaluated in the problem, the mechanical characteristics of the defined regions, nodal coordinates (x and y positions of the points in discretization), mesh topology, boundary conditions and number of crack extension increments. This archive is compatible with BEM programs and is generated in the ELASTOSTIC ANALYSIS module.

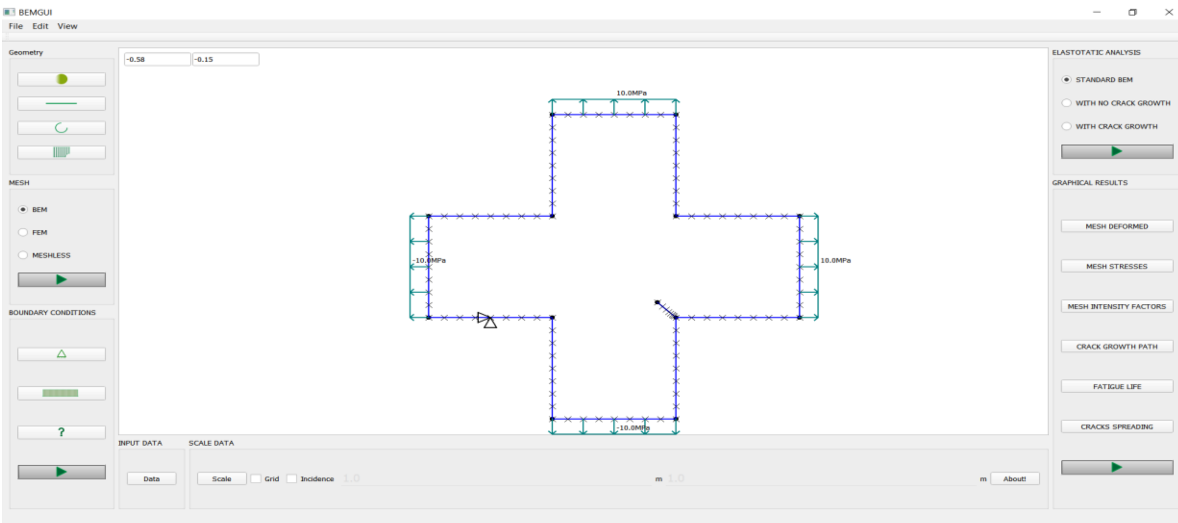

Figure 6: Final result of the modeling process

5. FINAL CONSIDERATIONS

The final result of the BEMGUI project offers a user-friendly, intuitive and fast interface. For greater user control, undo and redo functions (from the keyboard buttons “ctrl + Z” and “ctrl + Y”) have been added at all stages of the modeling process, as well as a way to exit a function by pressing the escape key. As stated earlier, the project builds on the object-oriented paradigm, which makes it easier to add new code snippets and reuse the implemented methods without changing the original work.

Finally, this project aims to be a useful tool at teaching and learning activities, both at graduate and, most importantly, post-graduate level. Its intention is to bring BEM closer to students and researches of the area and also make this efficient method easier to use.

REFERENCES

[1] Sampaio, F. F. Modelagem dinâmica computacional e o processo de ensino-aprendizagem: algumas questões para reflexão. Rio de Janeiro: NCE, UFRJ, 1999

[2] Rondón, EOO, Murakami, AE, Sakaguti, ES, Modelagem Computacional para Produção e Pesquisa em Avicultura. Revista Brasileira de Ciência Avícola [en linea] 2002.

[3] Booch, G., “Object-Oriented Analysis and Design with Applications”, The Benjamin/Cumming Publishing Company, Inc., 1994.

[4] Gomes, G.; Delgado neto, A. Modeling and 2D Cracks View Using Traction Boundary Integral Equation. XXXVII Iberian Latin American Congress on Computational Methods in Engineering - CILAMCE, Brasília, 2016.

[5] Sanner, Michel F. *et al*. Python: a programming language for software integration and development. J Mol Graph Model, v. 17, n. 1, p. 57-61, 1999.

[6] Hemmendinger, David, "Object-oriented programming." Britannica Academic, Encyclopædia Britannica, 2019. Disponível em <academic-eb-britannica.ez54.periodicos.capes.gov.br/levels/collegiate/article/object-oriented-programming/443530>. Acesso em 31/07/2019.

[7] Python Software Foundation, disponível em <https://www.python.org/>. Acesso em 29/07/2019

[8] The Qt Company, disponível em <https://www.qt.io/company>. Acesso em 29/07/2019

[9] Neto, Alvaro MD; Guimaraes, Denise C.; Gomes, Gilberto. Bemlab2d: GUI Modeling, Visualization And Analysis With Boundary Element–An Application Inelastostatic Problems. Revista Interdisciplinar De Pesquisa Em Engenharia, V. 2, N. 27.