

USING TRAME TO BUILD A WEB-BASED GRAPHICAL USER INTERFACE FOR ROXIE

A PROJECT REPORT

Submitted for the

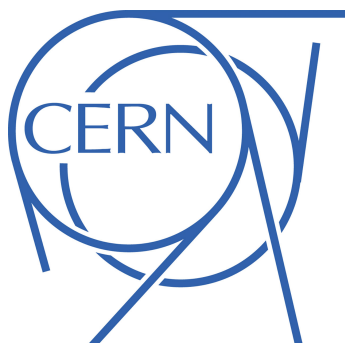
CERN SUMMER STUDENT PROGRAMME 2024

by

LUKE RASMUSSEN

Under the guidance of
Stephan Russenschuck

Matthias Bonora



TE-MS-C-TM

EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

1217 Meyrin, Switzerland

August 2024

Abstract

This report presents the successful development of a web-based Graphical User Interface (GUI) for the ROXIE software using the Trame framework. By leveraging Trame, the project aimed to simplify user interaction with ROXIE, making it more accessible to scientists and engineers. Key features of the interface include file management, data manipulation, and the visualization of 2D and 3D plots directly within a web browser. The report outlines the integration of Python with Vuetify for UI components, and the challenges encountered in adapting those components for the Trame environment. Future improvements and extensions are also discussed.

Acknowledgements

I would like to express my gratitude to Stephan Russenschuck and Matthias Bonora for selecting me as a Summer Student. Participation in this summer programme provided me with the valuable opportunity to learn about particle physics research and a variety of other interesting topics through the excellent workshops and lecture programme.

I was fortunate to work with some great colleagues in TE-MS-C-TM, with whom I enjoyed sharing daily conversations and participating in weekly science meetings. In particular, I am grateful to my direct supervisor Matthias Bonora for his help and guidance throughout my project. The experience of building a functioning project from scratch using unfamiliar tools has been very informative, and my skills and knowledge related to scientific software development have improved greatly thanks to this project.

Contents

Abstract	1
Acknowledgements	1
1 Introduction	3
2 Implementation	4
3 Conclusion	8

1 Introduction

The ROXIE program was created by Stephan Russenschuck at CERN for the electromagnetic simulation and optimization of accelerator magnets. ROXIE combines powerful geometry macros with the numerical accuracy of BEM-FEM coupling, hundreds of design variables and objective parameters, powerful optimization algorithms, and CAD/CAM interfaces.

Despite its powerful capabilities, ROXIE has a steep learning curve, partially due to its reliance on remote computing environments, which can be challenging to configure and operate. Consequently, there is a need for a more accessible interface that simplifies the process of running ROXIE.

This need could be addressed by developing a web-based Graphical User Interface (GUI) for ROXIE using the Trame framework. Trame is a Python-based framework designed for creating scientific web applications. Its primary advantages include:

- **Simplicity:** All logic and User Interface (UI) definitions are implemented in Python. This approach simplifies the development process by eliminating the need to juggle multiple programming languages and frameworks
- **Power:** Python's extensive ecosystem, including libraries for data visualization and processing like NumPy and Matplotlib, alongside integrations with powerful visualization tools such as VTK and ParaView, enables the creation of sophisticated and interactive scientific applications
- **Ubiquity:** The applications built with Trame can run on various platforms, including laptops, desktops, and cloud environments, with compatibility across different devices (phones, tablets, laptops, etc.)

The objective of this project was to explore Trame’s potential for developing a new GUI for ROXIE. This exploration involved:

- Building a GUI using Vue components within the Trame framework
- Writing Python code that interacts with the ROXIE API
- Demonstrating a proof of concept in the form of a functional web application

The proof of concept aimed to showcase the application’s effectiveness in performing simple tasks that are essential to scientists and engineers using ROXIE, particularly:

- Loading and saving data files
- Entering, editing, and deleting table data
- Displaying 2D plots
- Displaying 3D plots

2 Implementation

In the implementation of the web-based GUI, Trame was utilized as the overarching framework, which can integrate with Vue.js and Vuetify to build responsive and dynamic user interfaces.

Vuetify components, traditionally written in JavaScript, were translated into Python using Trame’s syntax. For example, Figure 1 illustrates the conversion of a VTooltip component, showcasing how JavaScript elements can be re-implemented in Python through Trame.

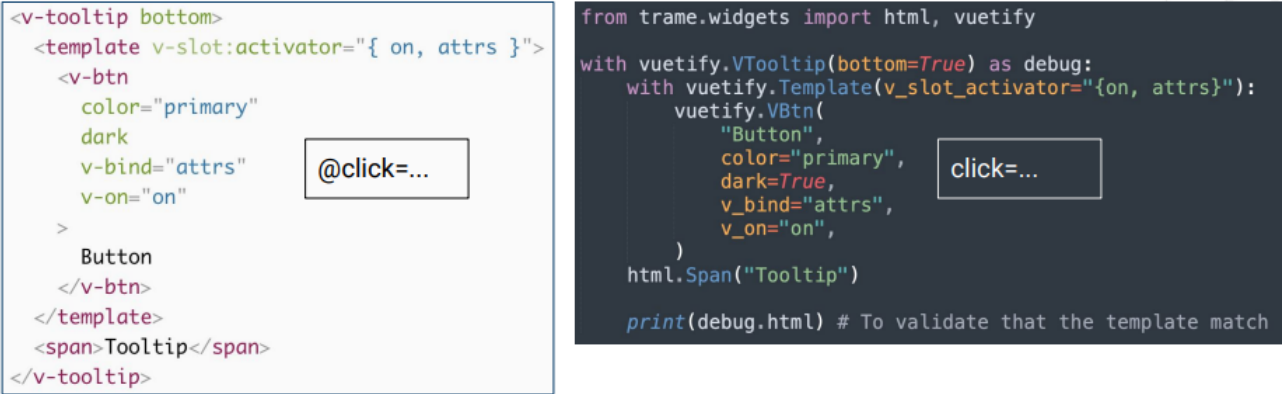


Figure 1: Converting a VTooltip component to Python

Despite the flexibility of the Vuetify framework, there was a learning curve in adapting it to work within Trame. While many existing Vuetify components worked well for specific actions, converting them from JavaScript to Python was sometimes a challenge, particularly when dealing with some aspects of Vuetify that are not directly available in Trame. This was especially true for more complex UI components, for example data tables with CRUD (create, read, update, delete) actions. Therefore, a significant part of the project involved testing and optimizing various UI components while working within the constraints of the Trame framework.

The entire codebase for the application is contained within a single Python script. By running this script, users can easily launch the GUI through a web browser without needing to manage multiple files. This approach also makes it easier to debug, update, and extend the application as needed.

Figure 2 shows the appearance of the GUI before any data has been loaded or entered. The interface includes a retractable drawer component on the left-hand side, which houses solid blue buttons that toggle the visibility of check boxes and table components. The tables within the GUI are feature-rich, supporting column sorting, the addition of new items, editing of existing entries, deletion of multiple items, and the adjustment of visible rows.

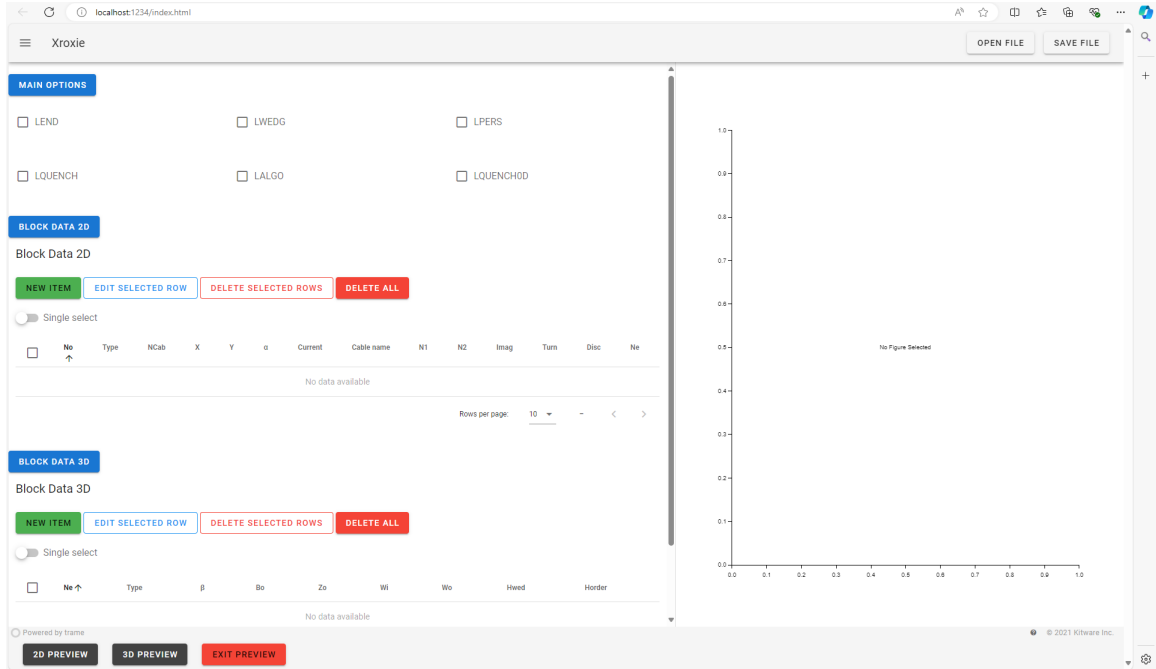


Figure 2: GUI with no data loaded

The toolbar located at the top of the interface includes buttons for saving data inputs to a file and opening existing data files. When a user opens a file, the data is parsed and populated into the relevant tables and check boxes within the GUI.

The footer of the interface is equipped with buttons for generating 2D and 3D plots, as well as for exiting these plots. These buttons trigger the ROXIE API directly from within the Python script, processing the input data and rendering the corresponding plots within the web browser. Figures 3 and 4 illustrate examples of 2D and 3D plots generated from data files.

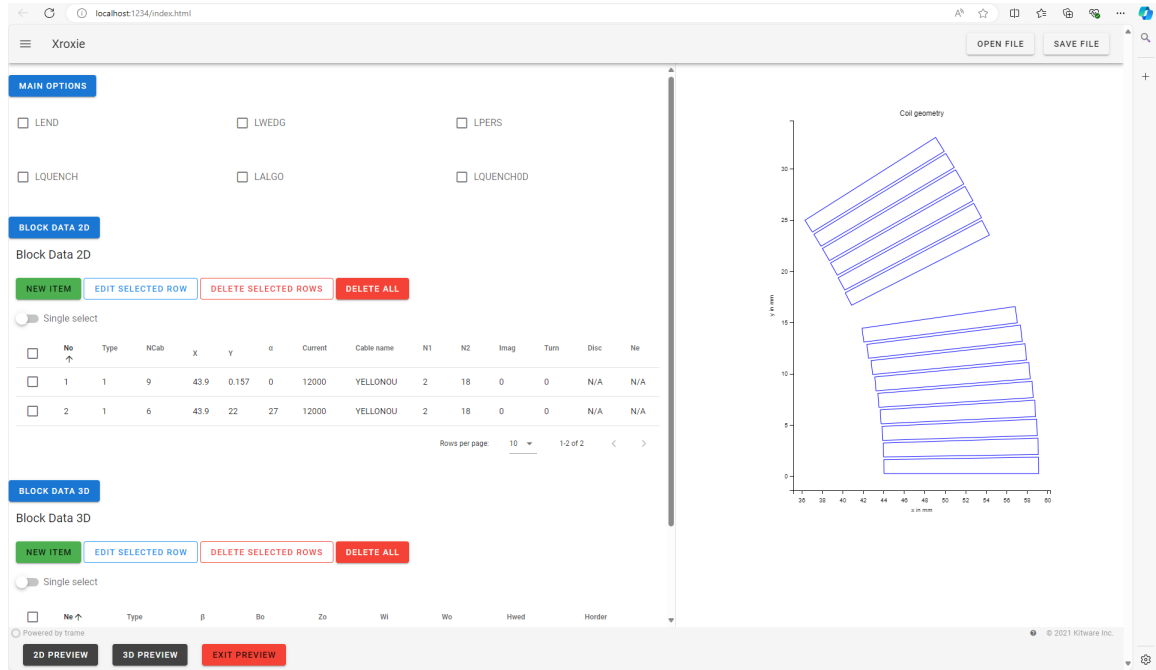


Figure 3: An example of a 2D plot generated from a datafile

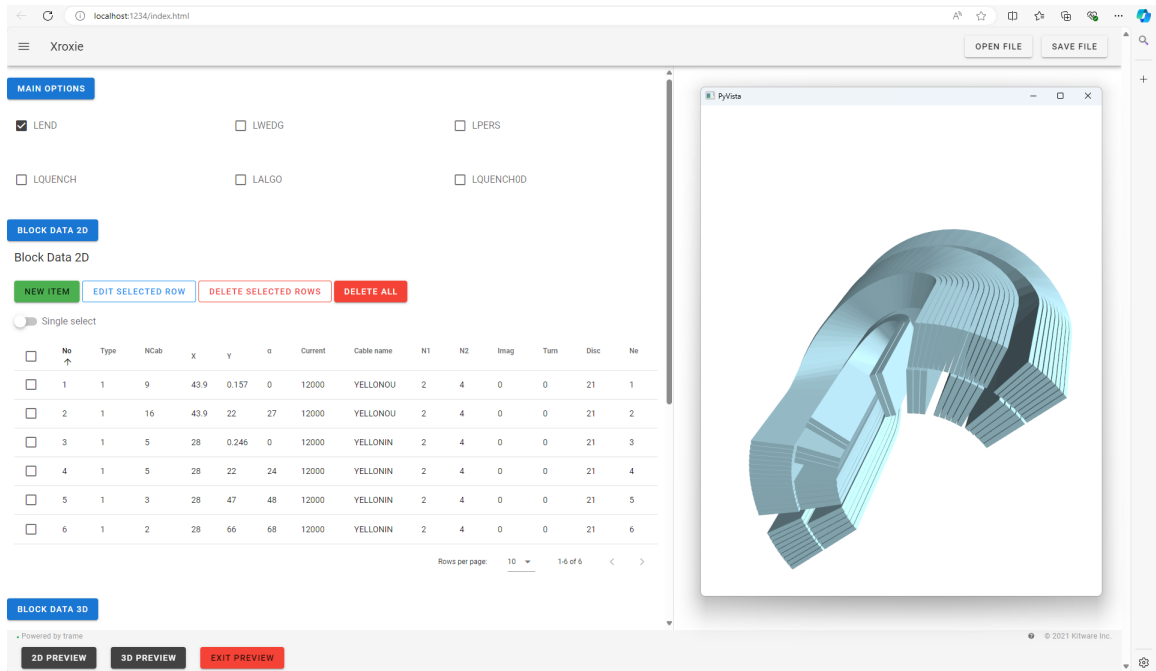


Figure 4: An example of a 3D plot generated from a datafile

3 Conclusion

This project successfully delivered a functional proof of concept for a ROXIE GUI that integrates essential features such as data file management, table manipulation, and 2D and 3D plotting, all within a web browser environment.

Overall, this project has laid a strong foundation for future enhancements to the web-based ROXIE GUI. Potential extensions could include the integration of more advanced functionalities, for example analytical and numerical field calculations, which would further augment its utility in scientific research and engineering design.