AE6102 - Parallel Scientific Computing and Visualization

Title - 3D Visualization and Analysis of Seismic Volumes

Final Project Report Spring 2023

Team & Participants

Team Name: **Sifar**

Name	Roll Number	Contact
Adarsh Raj	190050004	190050004@iitb.ac.in
Koustav Sen	190050062	190050062@iitb.ac.in
Raja Gond	190050096	190050096@iitb.ac.in

Abstract

The project provides a comprehensive and interactive visual representation of subsurface geology by creating three-dimensional images of seismic volumes in `MayaVI` library. The project also facilitates a better understanding of subsurface geology by allowing users to interact with the data in a more intuitive and efficient manner utilizing `TraitsUI` library. Visualization of seismic volumes is a very crucial component of interpretation workflows, be it to pick salt domes, interpret horizons, identify fault planes, or classify rock facies. For this, there are many specific visualization functions and geophysical functions implemented in same GUI using different modules.

Outline

Things We Did:

Our project involved developing a solution for 3D visualization and analysis of seismic volumes. Our main goal was to create a user-friendly GUI that could post-process segy data to numpy data, select labels and seismic data for a volume, and generate a 3D contour plot of the data with multiple functions for data visualization and analysis. Our solution was built using `Python` and several open-source libraries which are mentioned in the `Tools used` section above.

The GUI logical flow starts with choosing a seismic volume data and labels (if the data is associated with surface label data as well). The basic rendering of the data is a 3d Contour Plot which can be visualised in multiple viewpoints due to `MayaVi` robustness and features. There were some issues with the zooming functions of `MayaVi`, hence we implemented our own for user flexibility.

The key components for analysis and more detailed visualisation of the selected data are as follows:

- **Planar Plots:** This feature is used to generate planar plots for any plane of the volume, for any axes. This is controlled by the user input of slicing indices of inline (x-axis), crossline (y-axis) or depth (z-axis). The plots are generated using `matplotlib`.
- **Raw Data Analysis:** Another significant feature of our project is the ability to view raw data for any plane using the same input mechanics as above or coordinate (all three axes, for a single point). Since, seismic data can be of large size for a single plane and cannot be visualized properly on the GUI itself, we also added functionality to export it as CSV format. This feature allowed users to dig deeper into the data and perform more detailed analysis.
- **Animations:** We implemented animations using `mlab.animations`, that showed how the terrain changed for a certain axis, and users had the option to save those animations for later use. For saving purposes we used `ffmpeg` backend and `imageio` library.

- **Geophysical Analysis:** We also included a geophysical analysis feature that contained different seismic calculations using the `bruges` library. This feature allowed users to perform various calculations and analyses, including instantaneous phase, reflection strength (envelope), instantaneous frequency, etc. These plots are generally used by geophysicists to predict a types of terrains, existence of some specific geographical feature, etc.
- **Visualization Tools:** We included an extra visualisation for our data as a full 3d rendered object using `mayavi.pipeline`. We also added a volume analyser which is a great visualisation to understand the data, as it consists of multiple planes and each data point selected changes other plots and the 3d rendered object to cater according to that specific data point. This was done by an open source extension `volume_slice_analyser` of `mayavi`.

Did everything go as you planned and mentioned in the final proposal?

Yes, everything went as we planned in the project proposal. We successfully built the GUI application we mentioned and described in the final project proposal. In fact, we also made some advancements and added multiple features that we didn't plan earlier, like planar animations and raw data analysis. Also the ability to export data and save animations were also enhancements that we added after completing our initial project proposal requirements.

Did you encounter difficulties that you didn't anticipate? How did you overcome those?

Yes, we encountered a bunch of difficulties that we didn't anticipate. The prominent difficulties are as follows along with how we overcame those:

- **MayaVi Zoom Issue:** We were having trouble to zoom in and out using MayaVi default integration and struggled to find a solution in its documentation. We resolved this by creating our own buttons for the same.
- **Traits Class Embed in Another Traits Class:** We tried to embed multiple Traits class so that we can have two functionality, say input and output in the same GUI, but that was giving an error while using `PyQT5` backend. We resolved this issue by changing the backend toolkit to `qt4` and also allowed imports to `ETS_QT4`.
- **File Handling In TraitsUI:** The file select of TraitsUI was not working for our case even after we tried docs and several solutions available online. Lastly, we used file select from `tkinter` library to resolve this issue.
- **Segy Parsing:** We troubled during the parsing of segy files at first, because of its unique and complex structure, then we find the module `segyio`, which resolved the issue for us.

Tools we used in our project

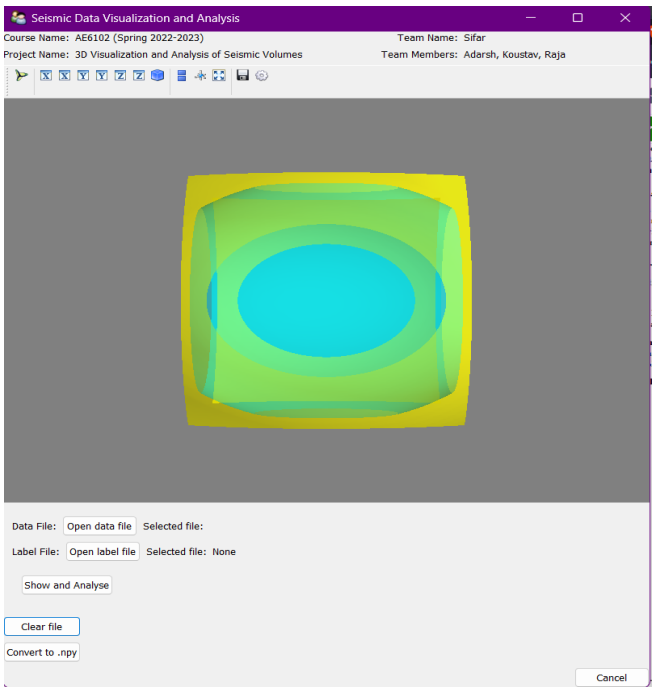
Tools we used and how we used them

Tool Name	Purpose
Segyio	A Python library for reading and writing SEG-Y formatted seismic data. SEG-Y is a commonly used file format in the geophysical industry for storing and exchanging seismic data. Segyio provides a convenient and efficient way to access and manipulate SEG-Y data in Python, including the ability to read and write SEG-Y files, access headers and trace data, and perform various processing tasks. The library is built on top of the numpy package, which allows for efficient data manipulation and processing.
Mayavi embedded in TraitsUI	TraitsUI is a powerful open-source library that provides a toolkit-independent GUI abstraction layer, which is used to support the "visualization" features of the Traits package. We can write a model using the Traits API and specify a GUI using the TraitsUI API (views, items, editors, etc.), and let TraitsUI and your selected toolkit back-end (Qt or Wx) take care of the details of displaying them. Mayavi is a Python package that provides 3D scientific data visualization capabilities. It is built on top of the Visualization Toolkit (VTK) and provides an easy-to-use interface for creating complex 3D visualizations of scientific data. We have used Mayavi embedded in traitsui to visualise 3d seismic volumes and analysis their different features.
NumPy and automan	In this project, we have processed SEG-Y seismic data by converting it into a 3D NumPy array, which has allowed us to apply various data processing techniques to the seismic data. We then visualized the results using Mayavi, a Python package for 3D scientific data visualization that is built on top of the Visualization Toolkit (VTK). To automate the conversion of SEG-Y files to NumPy format, we have used Automan, a simple automation framework that allows us to quickly and easily convert all the SEG-Y files present in a directory or a single SEG-Y file to a .npy file, which contains the required data in NumPy format.
Matplotlib	To plot analysis graph in traitsui window
imageio and ffmpeg	To save the animation generated
Sphinx	To generate Python documentation automatically. Our initial plan was to use Sphinx to generate complete documentation automatically. However, we had to modify our approach since a report was required as the final submission. As a result, we were only able to partially complete the automatic documentation generation using Sphinx.
Bruges	It's just a load of functions that implement important equations in (mostly seismic) geophysics. We have used this Python module to analyze the geophysical traits of our seismic volumes

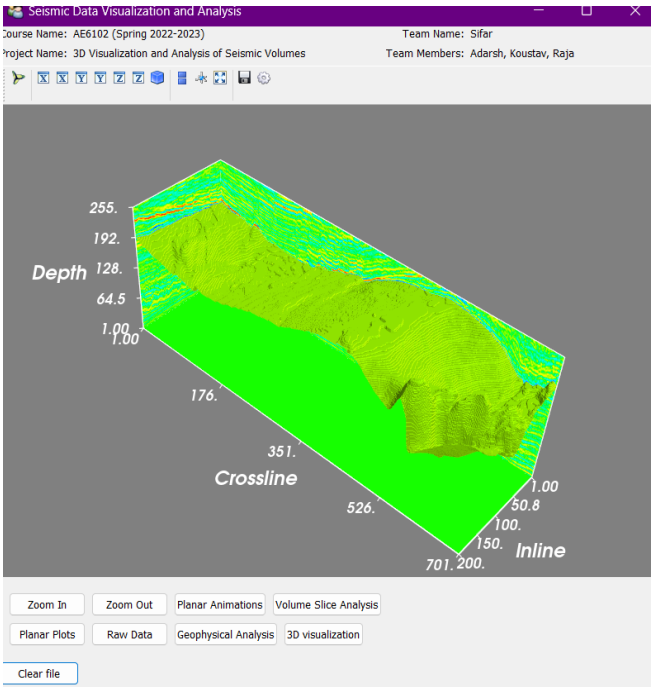
Please see requirements.txt for the detailed list of tools and packages we have used in our project along with their version number.

Showcase

These are some of the main application pages of our project, attached here with their description.

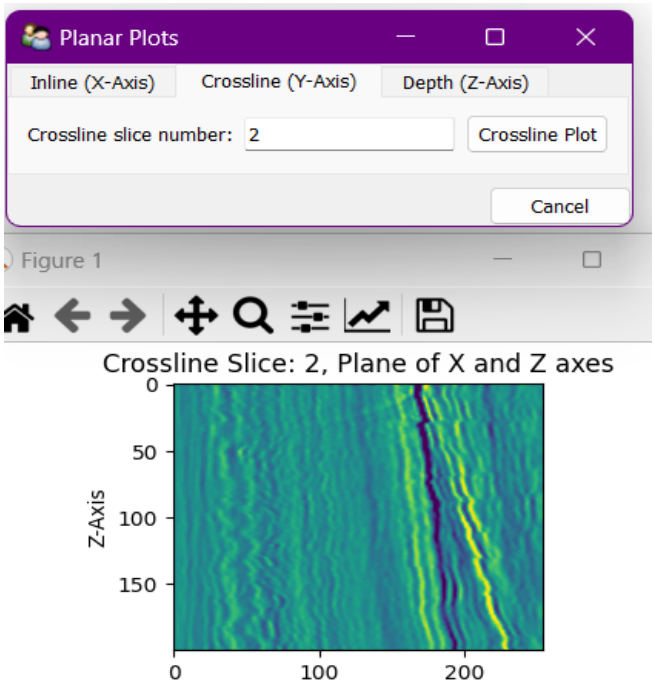


(a) Starting UI Page-> Upload Seismic Data Files for Visualization and Analysis

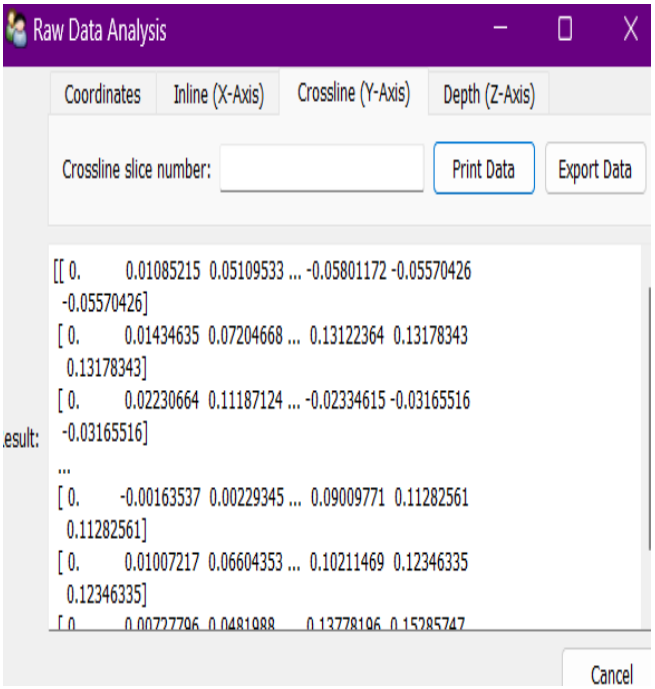


(b) After File Upload-> 3D Contour with different functions

Figure 1: Main Pages

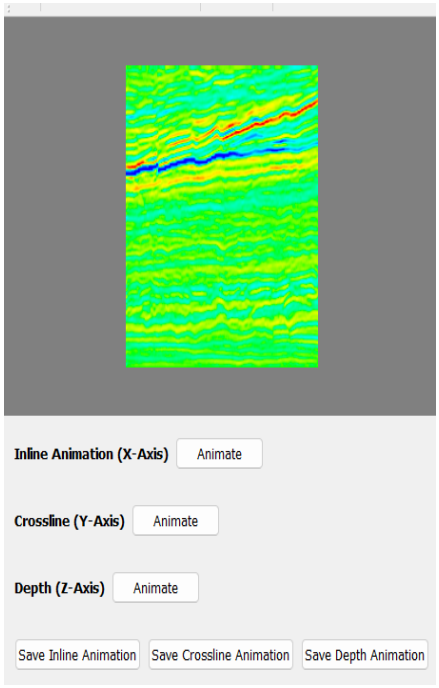


(a) Planar Plots UI

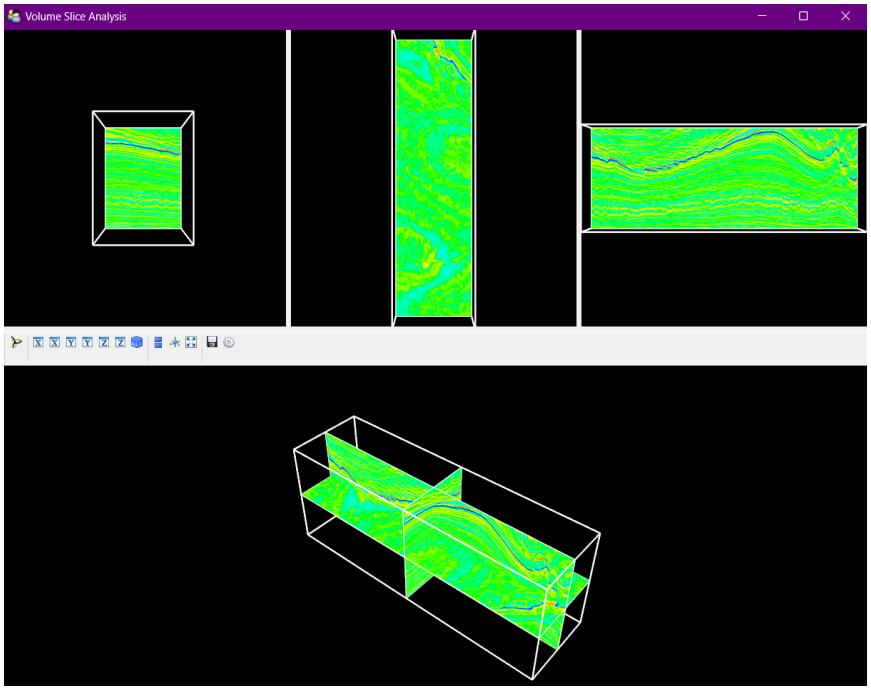


(b) Raw Data Analysis UI with Export Button

Figure 2: Analysis Functions I

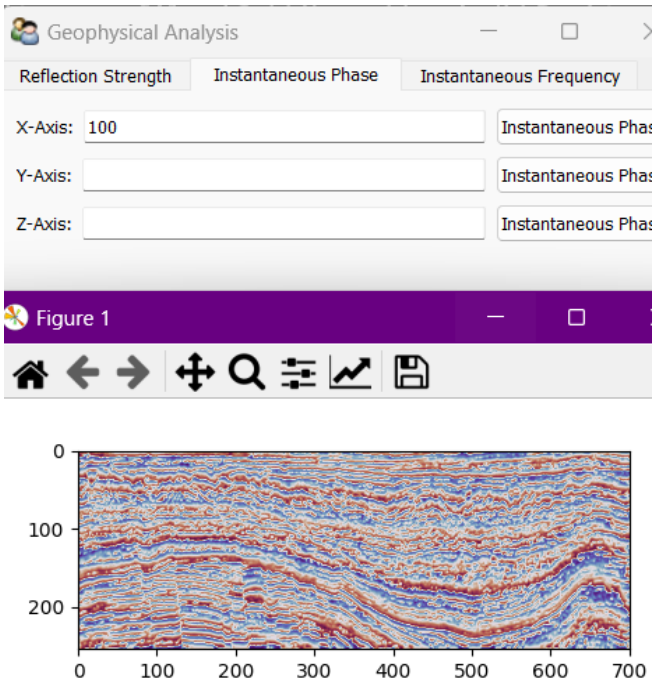


(a) Animations Save and View UI

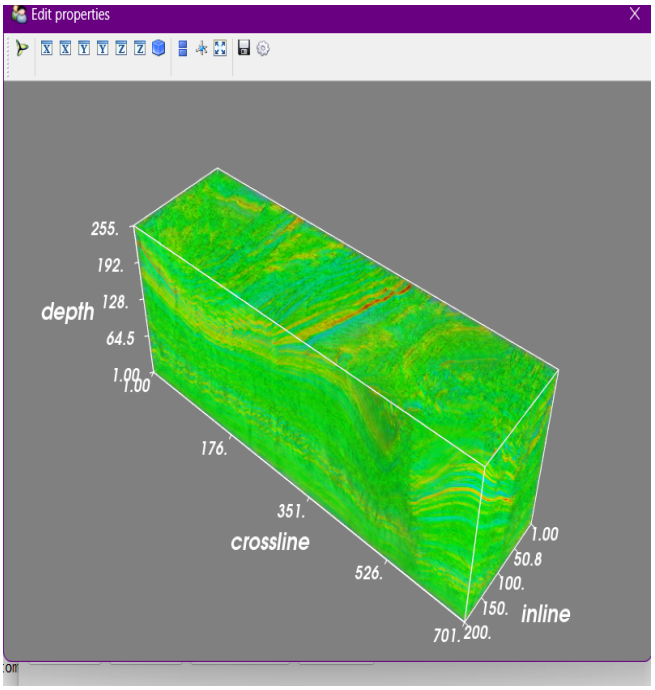


(b) Volume Slice Analysis of Seismic Data

Figure 3: Analysis Functions II



(a) Geophysical Analysis Functions



(b) 3d Rendering of Seismic Data Visualisation

Figure 4: Analysis Functions III

Note that, these are only some of the UI pages shown here, each feature is explandable to cover all range of data, not just some specific part or some specific axes. Some of these UI are also dynamic like Volume Slicer and Raw Data Analysis, the output trait (below one), changes according to the actions of user and shows error their itself, not on console.

Deliverables & Achievements

Seismic data visualization can be a challenging task for beginners and non-specialists, especially when using professional tools like OpendTect and Petrel. Setting up a project from scratch in these software tools requires configuring various parameters before the seismic volume of interest can be visualized, which can be overwhelming for users. Additionally, when working on Data Science projects involving seismic volumes, dealing with different data types for processing and loading data for visualization can be inconvenient. Repeatedly converting seismic data between different data types at different processing stages can be time-consuming and inconvenient for users.

What did we finally achieve?

With the help of `MayaVI` and `traitsUI` learned in the course, we develop a GUI application that provides an interface for users to visualize and analyze 3D seismic volumes easily. The application is designed to be simple and easy setup, making it accessible for beginners and non-specialists who may find professional tools like OpendTect and Petrel overwhelming.

How does our solution stand apart from similar solutions others?

Compared to similar solutions, your application stands apart in that it offers a more streamlined and customizable approach to seismic data visualization. The `Mayavi` library provided a powerful and flexible 3D visualization engine, while `TraitsUI` allowed for the creation of a user-friendly GUI that offers a range of interactive features for data exploration and analysis.

Of course, professional tools are a lot better than our application and they provide many more features/functionalities as compared to our project application.

How close were you able to get in terms of performance, or any other grounds for comparison that you would like to highlight?

In terms of performance, while our application may not have the full range of features and capabilities as professional tools, it offers a convenient and efficient solution for basic seismic data visualization and analysis. Additionally, the use of Python as the programming language and the open-source nature of the tools used allows for easy modification and customization to suit the specific needs of users.

What aspects of the course did you put to use in the project?

Overall, the knowledge and skills gained from the 3D visualization tools part of the course were essential in helping us to complete this project. The course provided a solid foundation in 3D visualization and Python programming, which were critical components in the development of this project.

Code

We used GitHub for the code repository and contributions to the project. We have pushed all the source code, test code and data pre-processing code in the same repository. We also have a corresponding `README` for installing the required modules and running our project on any other system.

Code Repository Link: `https://github.com/rajagond/AE6102_sifar`

Logistics

From - To (Date-Date)	Planned Progress	Status
12/02/2023 -18/02/2023	Project proposal(Draft) submission (Submitted on 18/02/2023)	Done
19/02/2023 -12/03/2023	Finalize project based on feedback received (Submission due on 20/03/2023)	Done
01/03/2023 -20/03/2023	Datasets Research, Data Collection, Research on Surface Geology for Analysis Mechanisms, Data Parsing and Transformation into `numpy` 3D models (Update-1 due on 20/03/2023)	Done
21/03/2023 -03/04/2023	We tested the `TraitsUI` library for GUI for a simple scene case of `MayaVI`. We spent most of our time during this period understanding how `MayaVI` and `TraitsUI` work because these tools are the backbone of our project and it was not covered in class at that time (Update-2 due on 03/04/2023)	Done
04/03/2023 -14/04/2023	We did logic building during this period for different features. For the code part, we implemented a UI interface to open data files and show its 3D visualisation (Update-3 due on 14/04/2023)	Done
15/04/2023 -03/04/2023	Final report and optimized code as an open-source GitHub repository. Almost all functionalities implementation is done during this period. Detailed description can be found in outline section (Final report due on 03/05/2023)	Done
04/05/2023 - 04/05/2023	Final presentation with TAs and instructor	-

Work distribution can be found in contributions section.

Miscellaneous

This project is aimed to create a small lightweight tool to analyse seismic volumes using 3d rendering and other traits. One can read the following articles or blogs to understand the `segy` data which is being analysed, and then the geophysical analysis and the cross axes plots will make much more sense to an external user.

- `https://agilescientific.com/blog/2014/3/26/what-is-seg-y.html`
- `https://help.dugeo.com/m/learning/l/1002051-the-dug-insight-guide-to-seg-y`
- Working with 3D seismic data

Contributions

The code was fairly distributed between Adarsh Raj and Raja Gond. Koustav Sen helped in building some parts of this report. For more detailed contributions specifics, please refer to the github commits.

References

- `https://wiki.seg.org/wiki/Open_data`
- `http://article.nadiapub.com/IJSIP/vol9_no5/39.pdf`
- `https://github.com/equinor/segyio`