3D Shear Force and Bending Moment Visualization using Xarray and Python

---

1. Introduction

This project focuses on the visualization of shear force and bending moment diagrams for a bridge grillage model using data provided in an Xarray dataset. The objective of the screening task was to generate both two-dimensional (2D) and three-dimensional (3D) visualizations of internal forces using Python libraries, following conventions used in structural post-processing tools such as MIDAS. The work demonstrates structured data extraction, continuity handling, and spatial visualization of structural response.

---

2. Dataset Description

The internal force data is provided in the form of an Xarray dataset, where element-wise results are stored along a dimension named *Component*. Each component corresponds to physical quantities such as bending moments (Mz_i, Mz_j) and shear forces (Vy_i, Vy_j) at the i-end and j-end of each element.

This structure allows efficient selection of required force components while preserving the original sign convention and element ordering defined in the dataset.

---

3. Task-1: 2D Shear Force and Bending Moment Diagrams

For Task-1, the central longitudinal girder was selected using the following element list:

15, 24, 33, 42, 51, 60, 69, 78, 83

Bending moment and shear force values were extracted using:

- Mz_i and Mz_j for the Bending Moment Diagram (BMD)
- Vy_i and Vy_j for the Shear Force Diagram (SFD)

The diagrams were constructed by stitching element-end values in sequence. For adjacent elements, the j-end value of the preceding element and the i-end value of the succeeding element showed smooth variation, indicating physical continuity. Minor numerical differences were observed, which are expected in finite element post-processed results and do not represent actual structural discontinuities.

The resulting 2D plots were continuous, physically meaningful, and followed the sign convention provided in the dataset.

---

4. Task-2: 3D Shear Force and Bending Moment Diagrams

For Task-2, three-dimensional diagrams were generated for all longitudinal girders. Geometry and connectivity were obtained from the provided *node.py* and *element.py* files, which define node coordinates and element connectivity using Python dictionaries.

The bridge geometry was represented in the X–Z plane using node coordinates, while internal force magnitudes were extruded along the Y-axis. This approach closely follows MIDAS-style post-processing, where force magnitudes are visualized as vertical offsets over the structural layout.

The base bridge geometry was plotted using all members, while force extrusion was applied only to the longitudinal girders. Separate 3D visualizations were produced for bending moment and shear force, resulting in clear and interpretable structural response plots.

---

5. Key Learnings and Reflections

This project reinforced the importance of understanding data structure before visualization. A key learning was distinguishing between Xarray coordinates and actual data variables, particularly when selecting force components stored along a dimension rather than as separate variables.

Another important takeaway was handling continuity in finite element outputs. Instead of forcing numerical equality at element joints, understanding physical continuity through smooth trends proved to be the correct approach.

Additionally, working with 3D plots highlighted the need to preserve physical meaning in axes rather than using categorical labels, which is crucial for professional structural visualization.

Overall, the project improved practical understanding of scientific data handling, structural visualization, and debugging workflows in Python.

---

6. Conclusion

The project successfully demonstrates the generation of 2D and 3D shear force and bending moment diagrams using Xarray and Python visualization tools. By combining structured data extraction with geometry-based visualization, the results accurately represent internal force distributions in a bridge grillage model. The approach is robust, transparent, and aligned with standard engineering post-processing practices.