GEO Matrix Transformation Visualizer v2.1
Graphics Project

May 2025

1 Project Description

The **GEO Matrix Transformation Visualizer v2.1** is a sophisticated Python-based graphical user interface (GUI) application developed to illustrate and explore 2D and 3D linear transformations through dynamic, animated visualizations.
This educational tool leverages **Tkinter** to create an intuitive and interactive user interface, while **Manim**, a powerful animation engine, is employed to render mathematically precise animations that vividly depict how matrices manipulate vector spaces and individual points.
The primary objective of this project is to serve as an effective learning resource for students and educators in the field of linear algebra, providing an engaging and visual approach to understanding complex mathematical concepts.

The application boasts a comprehensive set of features designed to enhance the learning experience:

- **Flexible Matrix Input**: Users can input 2x2 or 3x3 matrices through the Tkinter GUI, allowing for the exploration of various transformation types such as scaling, rotation, and shearing, depending on the matrix values entered.
- **Basis Vector Visualization**: The tool animates the transformation of basis vectors (î, ĵ, k), offering a clear representation of how the coordinate system itself is altered by matrix operations, which is fundamental to grasping linear transformations.
- **Custom Point Transformation**: It provides step-by-step animations of user-defined points undergoing transformation, breaking down the process into intermediate stages to highlight the effect of each matrix application.
- **Automated Video Output**: The application automatically generates and plays back video renderings of the transformations, enabling users to review and analyze the animations at their convenience.

To further enrich the submission, a 33-second demonstration video, `GEO_Visualizer_Demo.mp4`, is included, offering a concise yet comprehensive showcase of the GUI's interactivity and the quality of the animation output.
This application stands out as an invaluable resource for students of linear algebra, delivering interactive visualizations that clarify abstract concepts such as matrix multiplication, determinant effects, and vector space mappings.
The latest version, 2.1, introduces significant improvements, including enhanced 3D visualization capabilities with optimized camera angles for better depth perception, robust error handling to manage invalid inputs or missing dependencies gracefully, and the addition of custom point transformation animations that allow users to track specific points through sequential matrix applications, making it an even more powerful educational tool.

2 Screenshots

Below are screenshots showcasing the application's interface and outputs

Note: Please view `GEO_Visualizer_Demo.mp4` for a complete demonstration of the GUI and animation output.

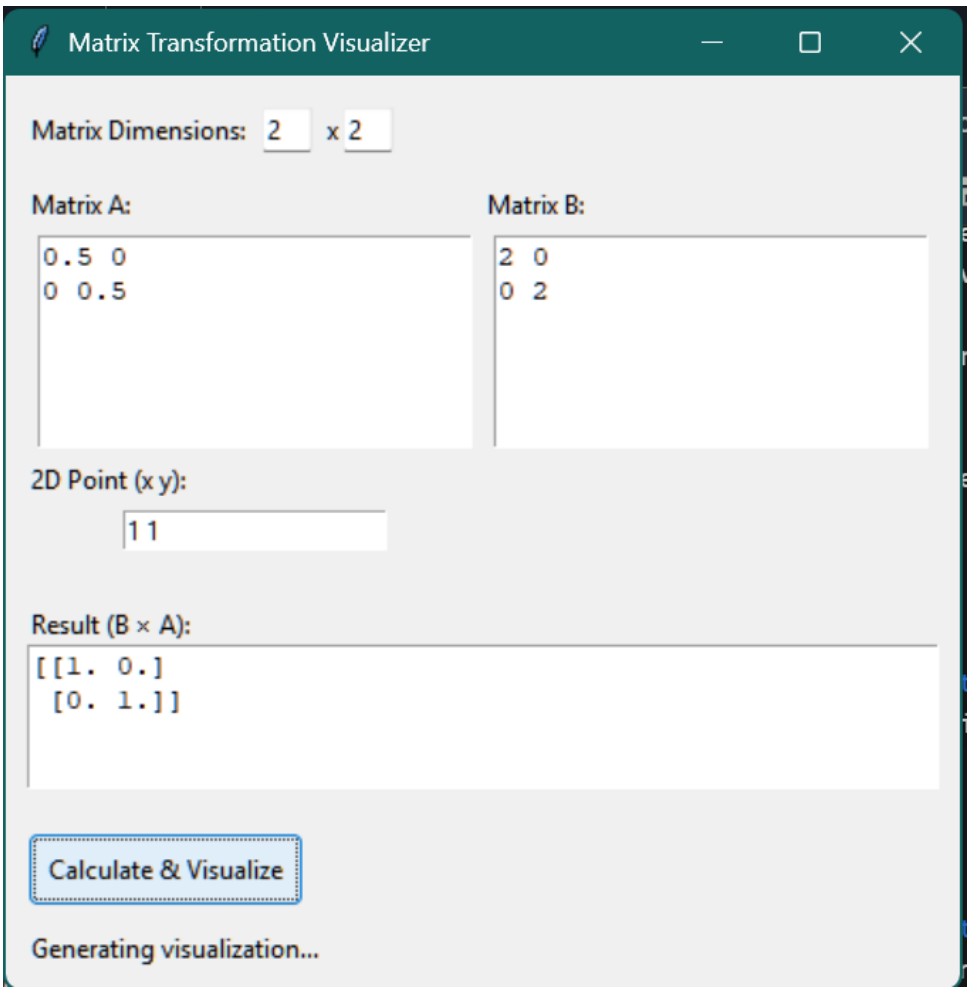

Figure 1: Tkinter GUI with matrix and 2D point input fields.

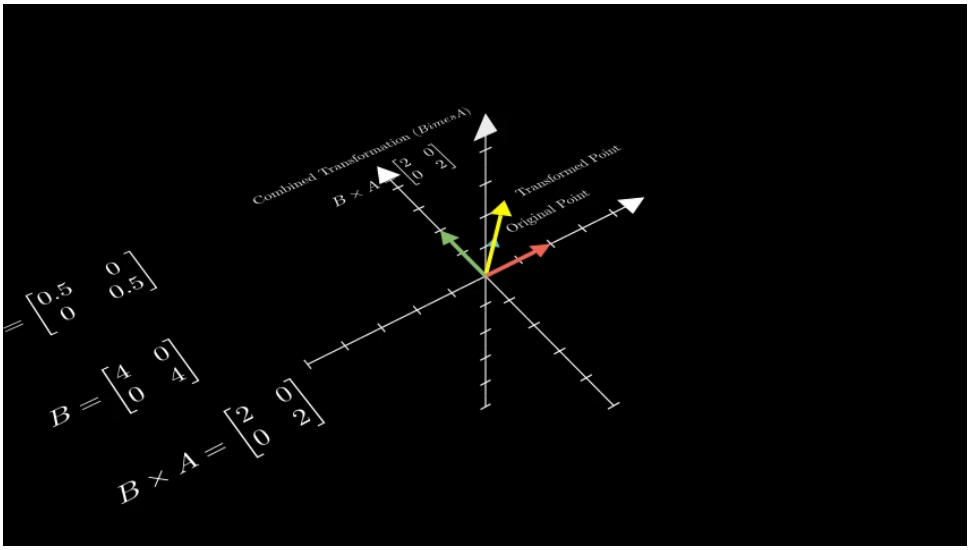

Figure 2: 2D transformation with basis vectors and custom point.

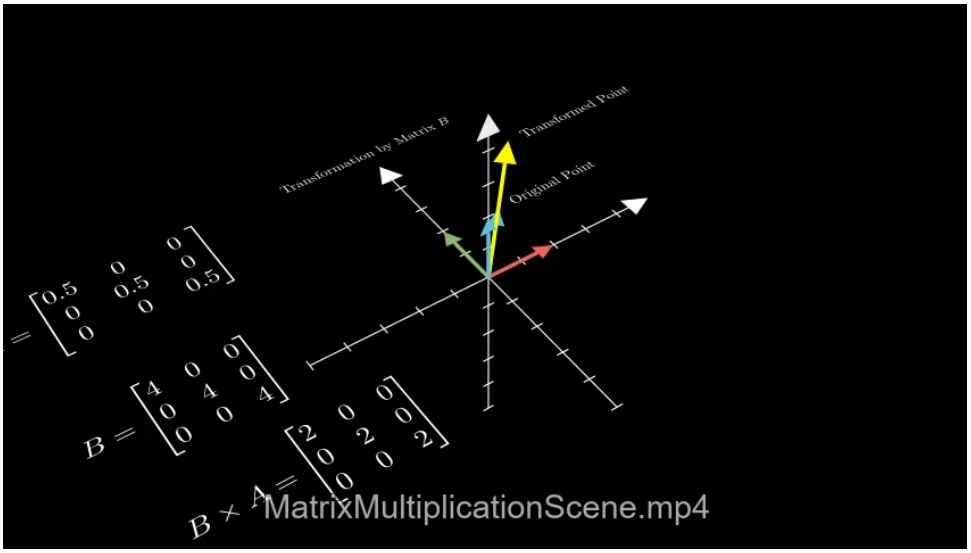

Figure 3: 3D transformation with basis vectors and custom point.

3 Code

The core functionality resides in `main_gui.py`. Below is a key snippet demonstrating matrix input and animation triggering (full code available in the attachments):

```
def parse_matrix(matrix_str, rows, cols):
    """Parse input text to create a numpy matrix"""
    try:
        matrix = []
        lines = [line.strip() for line in matrix_str.strip().split('\n') if line.strip()]

        if len(lines) != rows:
            raise ValueError(f"Expected {rows} rows, got {len(lines)}")

        for line in lines:
            elements = [float(num) for num in line.split()]
            if len(elements) != cols:
                raise ValueError(f"Expected {cols} columns in each row, got {len(elements)}")
            matrix.append(elements)

        return np.array(matrix)
    except Exception as e:
        raise ValueError(f"Error parsing matrix: {e}")
```

The application dynamically generates `matrix_visualization.py` for Manim animations, with dependency checks handled by `check_environment.py`.

4 Setup Instructions

To run `main_gui.py`:

1. Install Python 3.8+, FFmpeg, and MiKTeX (Windows) or equivalent LaTeX distribution.
2. Run `setup_environment.bat` (Windows) or `setup_environment.sh` (Mac/Linux).
3. Install dependencies: `pip install manim numpy`.
4. Execute: `python main_gui.py`.

5 Conclusion

The GEO Matrix Transformation Visualizer v2.1 successfully providing an intuitive and educational tool for visualizing linear transformations.
The included video (`GEO_Visualizer_Demo.mp4`) enhances the submission by demonstrating the application in action.
Future improvements could include real-time previews and additional transformation types.