MATRIX & CALCULUS OPERATIONS CALCULATOR

112003133

Shlok Deshpande

S3

Rapid Prototyping Practice Using Object Oriented Programming Project

INTRODUCTION

The project is based on the **problem statement :**

1. Create an application with a Graphical User Interface (GUI) to provide an integrated operations and solutions calculator to solve the questions of students relating the field of mathematics with an interactive interface.
2. The project utilises the concepts of matrices and calculus to provide accurate results on the operations on the matrices and functions(x,y) given by the user and display the results, all integrated in a graphical user interface.
3. The application provides support for 12 Distinct Operations in Matrices and Calculus : **"Matrix Multiplication", "Matrix Addition", "Determinant", "Inverse", "Transpose", "Power of", "Matrix Subtraction", "Differentiation", "Integration", "Double Integration", "Multiple Differentiation", "Roots of Polynomial".**

CONCEPTS & LIBRARIES USED

Brief Overview of the concepts of maths and the libraries of Python used to create this project :

1. Matrix Operations Concepts
2. Object Oriented Programming Concepts
3. Tkinter Library (GUI with Python)
4. SymPy Library (Calculus Solutions Library for Python)
5. NumPy Library (Roots of Polynomial)

PYTHON LIBRARIES IN DETAIL

Tkinter Library :

1. Tkinter is the standard **GUI** library for Python. Python when combined with Tkinter provides a fast and easy way to create **GUI** applications. Tkinter provides a powerful object-oriented interface to the Tk **GUI** toolkit. In fact, it's the only framework built into the Python standard library.
2. **Widgets** : Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets. Each widgets have specified and common attributes (ex. size, bg, font etc.).
3. **Geometry Management** : All Tkinter widgets have access to specific geometry management methods, which have the purpose of organizing widgets throughout the parent widget area. Tkinter exposes the following geometry manager classes: pack, grid, and place.

PYTHON LIBRARIES IN DETAIL

SymPy Library :

1. It is a Python library for symbolic mathematics. It aims to become a full-featured computer algebra system (CAS) while keeping the code as simple as possible in order to be comprehensible and easily extensible. **SymPy** is written entirely in Python.
2. It only depends on **mpmath**, a pure Python library for arbitrary floating point arithmetic, making it easy to use.
3. **SymPy** can be used for the following tasks and much more :
4. Evaluate expressions with arbitrary precision. Perform algebraic manipulations on symbolic expressions. Perform basic calculus tasks (limits, differentiation and integration) with symbolic expressions.

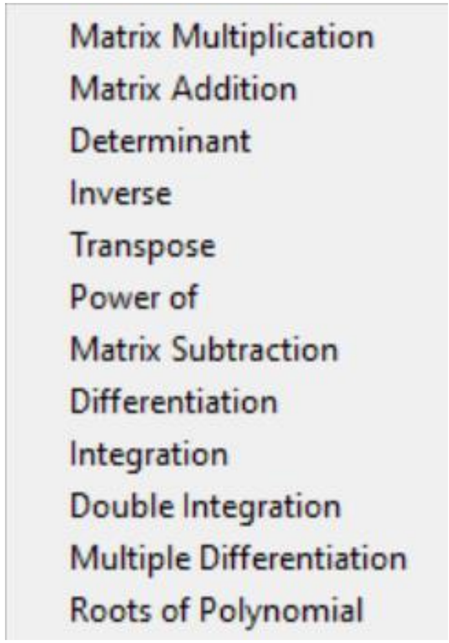
NumPy Library (Polynomial Module):

1. This module provides a number of objects (mostly functions) useful for dealing with polynomials, including a **Polynomial** class that encapsulates the usual arithmetic operations.

USER INTERFACE & FEATURES

Basic Overview of the features and user interface :

1. These are the 12 modes in the selection panel in the GUI application.
2. **Matrix Mul/Add/Sub** : You are asked to input no. of rows (n) and no. of columns (m) of the 2 matrices you want to operate on. n & m can be any number. Then you will get 2 input matrices on the screen where you can input values and calculate the results.
3. **Matrix Det/Inv/Pow of/ Transpose** : You are asked to input no. of rows (n) and no. of columns (m) of the matrix you want to operate on. Then you will get an (nxm) input matrix on the screen where you can input values and calculate the results.
4. **Diff/Int** : You are asked to input the eqn f(x,y) that you want to operate on and limits in case you want to find value of definite integral, else you will get function obtained after indefinite integration.
5. **Roots of Polynomial** : You are asked to input order of polynomial and then the coefficients of the polynomial. The roots of the equation will then be displayed below.



WORKING OF THE PROJECT

matrix_classes.py

The file contains the class **Matrix** which has the functions for each operation on the matrix and each specific function is called when we want to display the result of that operation on the matrices given by the user. It also contains the class **Polynomial** which has the function to evaluate the roots of the polynomial using **NumPy**.

matrix_ui.py

The main file, consist of the classes SimpleTableInput, Example1, Example2 and the respective functions for the Differentiation & Integration modes.

1. **SimpleTableInput –** Class that creates and gets the input nxm matrix from the user.
2. **PolynomialInput –** Class that creates and get the input coefficients for the n order polynomial from the user.
3. **Example 1/Example 2 /Example 3 –** Classes needed to generate the buttons, labels and table inputs for a particular frame, and display the result. Divided based on whether single or double matrix input is required or the polynomial mode was chosen. They support multiple operations on same matrix. They call **Matrix & Polynomial** class function to get output.
4. The rest of program consist of essential functions and the main loop of the Tkinter GUI which generates the row/column input and selection list.

FUTURE SCOPE

1. The application can be enhanced by increasing the operations the app supports, we can also include calculators from other fields of mathematics (Linear Algebra, Vector Calculus etc.).
2. We can also deploy the proper application on the web using any of the numerous frameworks available.
3. The application can also be designed further to showcase the step by step solution to obtain the particular result for the benefit of the user.
4. Better usability and functionality of the application can also be obtained by reducing errors and creating a better GUI experience with less room for anticipation for the user using additional libraries of python.