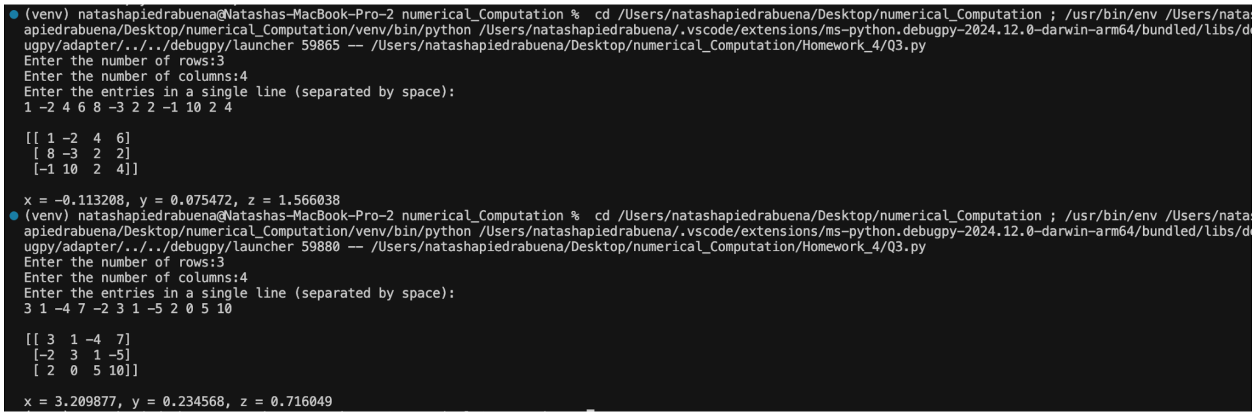
Natasha Piedrabuena
Numerical Computation
Cramer System of Linear Equation

In this project, I develop a computational approach to evaluate determinants and solve linear systems of equations using Cramer's Rule. Determinants play a crucial role in linear algebra, as they help us determine whether a system of linear equations has a unique solution. In this case, I used a built-in function provided by the numpy library, which simplifies determinant calculation significantly and allows for efficient handling of matrix operations [1].

For an n x m matrix, the determinant is a scalar value that provides insights into the matrix's properties—specifically, whether it is invertible. Using a recursive method, With this determinant function, we implement Cramer's Rule to solve a system of equations by constructing modified matrices for each variable. If the determinant of the original matrix is non-zero, the system has a unique solution, which we compute by evaluating the determinants of these modified matrices. This project demonstrates a practical application of determinants in solving linear systems efficiently.



```
● (venv) natashapiedrabuena@Natashas-MacBook-Pro-2 numerical_Computation %  cd /Users/natashapiedrabuena/Desktop/numerical_Computation ; /usr/bin/env /Users/nata
  apiedrabuena/Desktop/numerical_Computation/venv/bin/python /Users/natashapiedrabuena/.vscode/extensions/ms-python.debugpy-2024.12.0-darwin-arm64/bundled/libs/d
  ugpy/adapter/../../debugpy/launcher 59865 -- /Users/natashapiedrabuena/Desktop/numerical_Computation/Homework_4/Q3.py
  Enter the number of rows:3
  Enter the number of columns:4
  Enter the entries in a single line (separated by space):
  1 -2 4 6 8 -3 2 2 -1 10 2 4

  [[ 1 -2  4  6]
   [ 8 -3  2  2]
   [-1 10  2  4]]

  x = -0.113208, y = 0.075472, z = 1.566038
● (venv) natashapiedrabuena@Natashas-MacBook-Pro-2 numerical_Computation %  cd /Users/natashapiedrabuena/Desktop/numerical_Computation ; /usr/bin/env /Users/nata
  apiedrabuena/Desktop/numerical_Computation/venv/bin/python /Users/natashapiedrabuena/.vscode/extensions/ms-python.debugpy-2024.12.0-darwin-arm64/bundled/libs/d
  ugpy/adapter/../../debugpy/launcher 59880 -- /Users/natashapiedrabuena/Desktop/numerical_Computation/Homework_4/Q3.py
  Enter the number of rows:3
  Enter the number of columns:4
  Enter the entries in a single line (separated by space):
  3 1 -4 7 -2 3 1 -5 2 0 5 10

  [[ 3  1 -4  7]
   [-2  3  1 -5]
   [ 2  0  5 10]]

  x = 3.209877, y = 0.234568, z = 0.716049
```

For the second part of the code, the program outputs the initial matrix to verify the input before proceeding with the solution. This allows a user to check the structure and values of the input matrix. My code is designed to take integers for the number of rows and columns and then accepts values for each element in the matrix, separated by spaces. These values are processed and organized into a matrix form using numpy.array [2]. The program then reshapes this input into a matrix and calls the cramer function, defined in the Q1.py file, to execute the solution procedure. Each line of the code is commented to clarify its function, making the steps transparent and easier to understand. I verified the output by comparing it with online tools that solve linear equations using Cramer's Rule, and my results matched accurately with these external solutions.

This project illustrates the importance of determinants and Cramer's Rule play in solving linear systems of equations. By calculating determinants and using them to assess whether a system has a unique solution, this approach provides a structured and efficient way to address linear algebra problems. The code successfully automates this solution process, allowing users to input any n×nn×n matrix and receive solutions—if they exist. Through this project, I learned the value of matrix operations and determinants in linear algebra, and I also gained practical experience in coding and implementing mathematical rules. This application demonstrates how theoretical concepts in linear algebra, such as determinants and matrix inversions, can be translated into practical computational solutions.

[1]
GeeksforGeeks, "Take Matrix Input from User in Python," *GeeksforGeeks*, Aug. 3, 2020. [Online]. Available: https://www.geeksforgeeks.org/take-matrix-input-from-user-in-python/.

[2]
GeeksforGeeks, "How to Calculate the Determinant of a Matrix using NumPy," *GeeksforGeeks*, Sept. 5, 2020. [Online]. Available: https://www.geeksforgeeks.org/how-to-calculate-the-determinant-of-a-matrix-using-numpy/.