Assignment
Linear Algebra and Matrix Theory

July 16, 2025

Assignment Instructions

This assignment is designed to assess your understanding of the foundations of linear algebra and matrix theory. It includes a mix of theoretical and coding-based questions.

- **Theoretical questions:** These must be answered using your own understanding and words. The responses will be closely reviewed for originality and will be checked for any form of AI-generated content. Please explain concepts clearly and concisely.
- **NumPy-based questions:** For problems requiring computational solutions using NumPy, submit:
  - The `Python` code you used to solve the problem.
  - A screenshot or image of the output displayed in your terminal or notebook.
- The final document should be compiled as a **PDF file**, which includes:
  - A cover page
  - Table of contents
  - All answers in order
- **Deadline:** July 21, 2025

Good luck, and take this as an opportunity to strengthen your mathematical thinking and coding fluency.

Assignment: Linear Algebra and Matrix Theory

1. Define a linear transformation. Show whether the function $T(x, y) = (2x, 3y)$ is a linear transformation by checking the two required properties.

2. Write a NumPy function that checks if a given transformation matrix satisfies additivity and scalar multiplication preservation (i.e., linearity).

3. Classify the following system as consistent/inconsistent and dependent/independent:

$$\begin{cases} x + 2y = 3 \\ 2x + 4y = 6 \end{cases}$$

   Justify your answer using matrix rank.

4. Solve the system of equations using both NumPy's `np.linalg.solve` and `np.linalg.lstsq`. Compare the results:

$$\begin{cases} 2x + 3y = 8 \\ 5x + 4y = 13 \end{cases}$$

5. Given the matrix $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, compute the following using NumPy:

   - Transpose
   - Inverse
   - Determinant
   - Verify that $AA^{-1} \approx I$

6. Perform LU decomposition on a 3×3 matrix using SciPy. Interpret the resulting matrices L, U, P and describe their utility in solving linear systems.

7. Solve the following overdetermined system using QR decomposition:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

   Use NumPy to perform decomposition and back-substitution.

8. Explain the geometric interpretation of consistent and inconsistent linear systems. Create and solve one example of each using NumPy, then visualize in 2D using matplotlib.

9. Why is matrix invertibility important in solving linear systems? Give an example of a non-invertible matrix and interpret the result in terms of system solutions.

10. Write a Python script using NumPy that classifies a given system $AX = b$ as:

- Consistent with a unique solution
- Consistent with infinite solutions
- Inconsistent

11. Explain the difference between "basis of a vector space" and "basis of a column space" with a concrete example.

12. Use NumPy to check whether the following vectors form a basis for $\mathbb{R}^3$:

```
v1 = [1, 0, 0]
v2 = [0, 1, 0]
v3 = [0, 0, 1]
```

13. State and explain the rank-nullity theorem. Provide a matrix example with full explanation.

14. Compute the rank of the following matrix using NumPy:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 1 & 1 & 1 \end{bmatrix}$$

    Explain why the rank is less than 3.

15. Prove that the rank of a matrix equals the number of pivot columns in its row echelon form. Illustrate with an example matrix.

16. Construct a 3×3 matrix of rank 1. Use NumPy to verify that it has only one linearly independent column.

17. Write a Python function to generate 10 random 4×4 matrices. For each, compute its rank and determine how many are full rank. Report the percentage.

18. Prove that any n linearly independent vectors in $\mathbb{R}^n$ form a basis. Verify this numerically with three vectors in $\mathbb{R}^3$.

19. Consider the matrix:

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

    Find the rank of A and determine which columns form a basis for its column space.

20. Write a Python script that takes any matrix as input and outputs a set of linearly independent columns (i.e., a basis for the column space).