12.40

Harsha-EE25BTECH11026

September 12,2025

Question

Given $\mathbf{M} = \begin{pmatrix} 2 & 3 & 7 \\ 6 & 4 & 7 \\ 4 & 6 & 14 \end{pmatrix}$. Which of the following statements is/are correct:

1. The rank of $\mathbf{M}$ is 2
2. The rank of $\mathbf{M}$ is 3
3. The rows of $\mathbf{M}$ are linearly independent
4. The determinant of $\mathbf{M}$ is zero.

Theoretical Solution

Upon row reduction of matrix **M** to Row Echelon form (REF),

$$\begin{pmatrix} 2 & 3 & 7 \\ 6 & 4 & 7 \\ 4 & 6 & 14 \end{pmatrix} \xleftrightarrow[R_3 \leftarrow R_3 - 2 \times R_1]{R_2 \leftarrow R_2 - 3 \times R_1} \begin{pmatrix} 2 & 3 & 7 \\ 0 & -5 & -14 \\ 0 & 0 & 0 \end{pmatrix} \tag{1}$$

$\implies$ (a) The rank of **M** is 2

(b) The determinant of **M** is 0

C Code -Finding REF of a matrix

```c
#include <stdio.h>

#define ROWS 3
#define COLS 3

void row_echelon_form(double A[ROWS][COLS]) {
    int pivot_row = 0;

    for (int pivot_col = 0; pivot_col < COLS; pivot_col++) {
        int pivot = -1;
        for (int r = pivot_row; r < ROWS; r++) {
            if (A[r][pivot_col] != 0.0) {
                pivot = r;
                break;
            }
        }
        if (pivot == -1) continue;
```

C Code -Finding REF of a matrix

```
        if (pivot != pivot_row) {
            for (int c = 0; c < COLS; c++) {
                double tmp = A[pivot_row][c];
                A[pivot_row][c] = A[pivot][c];
                A[pivot][c] = tmp;
            }
        }
        for (int r = pivot_row + 1; r < ROWS; r++) {
            if (A[r][pivot_col] != 0.0) {
                double factor = A[r][pivot_col] / A[pivot_row][
                    pivot_col];
                for (int c = pivot_col; c < COLS; c++) {
                    A[r][c] -= factor * A[pivot_row][c];
                }
            }
        }pivot_row++;
        if (pivot_row == ROWS) break;
    }
}
```

C Code -Finding REF of a matrix

```c
void solve_ref(double *out) {
    double A[ROWS][COLS] = {
        {2, 3, 7},
        {6, 4, 7},
        {4, 6, 14}
    };

    row_echelon_form(A);

    int k = 0;
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
            out[k++] = A[i][j];
        }
    }
}
```

Python+C code

```python
import ctypes
import sympy as sp

# Load library
lib = ctypes.CDLL("./libref_solver.so")

# Prepare result buffer (3x3 = 9 doubles)
result = (ctypes.c_double * 9)()
lib.solve_ref(result)

# Convert to SymPy Matrix
ref = sp.Matrix(3, 3, result)
print("Row Echelon Form (REF):")
sp.pprint(ref)
```

Python code

```python
import sympy as sp

A=sp.Matrix([[2,3,7],[6,4,7],[4,6,14]])

ref_A=A.echelon_form()
print("Row Echelon Form:")
sp.pprint(ref_A)
```