Presentation - Matgeo

Aryansingh Sonaye
AI25BTECH11032
EE1030 - Matrix Theory

September 27, 2025

Problem Statement

Problem 5.13.18
If the system of linear equations

$$x + ky + 3z = 0, \tag{1.1}$$
$$3x + ky - 2z = 0, \tag{1.2}$$
$$2x + 4y - 3z = 0 \tag{1.3}$$

has a non-zero solution (x, y, z), then $\frac{xz}{y^2}$ is equal to

$$\text{a) } 10 \quad \text{b) } -30 \quad \text{c) } 30 \quad \text{d) } -10 \tag{1.4}$$

Description of Variables used

Variable	**Description**
x, y, z	Unknowns of the system
k	Parameter in the system

Theoretical Solution

Start with the augmented matrix:

$$\begin{pmatrix} 1 & k & 3 & 0 \\ 3 & k & -2 & 0 \\ 2 & 4 & -3 & 0 \end{pmatrix}. \tag{2.1}$$

Eliminating below the first pivot:

$$R_2 \to R_2 - 3R_1, \quad R_3 \to R_3 - 2R_1 \;\Rightarrow\; \begin{pmatrix} 1 & k & 3 & 0 \\ 0 & -2k & -11 & 0 \\ 0 & 4-2k & -9 & 0 \end{pmatrix}. \tag{2.2}$$

Next, remove the second entry in row 3:

$$R_3 \to R_3 + \left(\tfrac{2}{k} - 1\right) R_2 \;\Rightarrow\; \begin{pmatrix} 1 & k & 3 & 0 \\ 0 & -2k & -11 & 0 \\ 0 & 0 & \frac{2(k-11)}{k} & 0 \end{pmatrix}. \tag{2.3}$$

Theoretical Solution

For a homogeneous system $A\mathbf{v} = 0$, a non-trivial solution exists only if $\operatorname{rank}(A) < 3$. Hence the last pivot must vanish:

$$\frac{2(k-11)}{k} = 0 \;\Rightarrow\; k = 11. \tag{2.4}$$

Substitute $k = 11$:

$$\begin{pmatrix} 1 & 11 & 3 & 0 \\ 3 & 11 & -2 & 0 \\ 2 & 4 & -3 & 0 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 11 & 3 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{2.5}$$

From row 2: $2y + z = 0 \Rightarrow z = -2y$.
From row 1: $x + 11y + 3z = 0 \Rightarrow x = -5y$.

Theoretical Solution

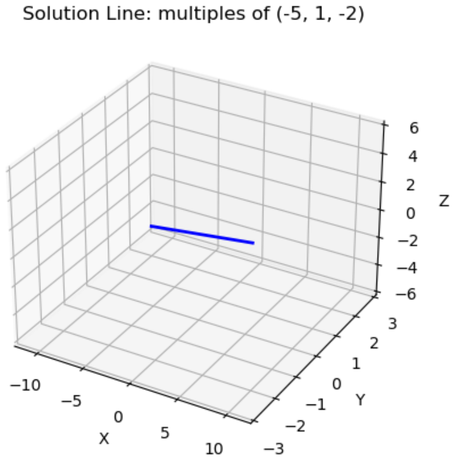
$$\mathbf{v} = y \begin{pmatrix} -5 \\ 1 \\ -2 \end{pmatrix}, \quad y \neq 0. \tag{2.6}$$

Finally,

$$\frac{xz}{y^2} = \frac{(-5y)(-2y)}{y^2} = 10. \tag{2.7}$$

$$\boxed{10} \tag{2.8}$$

Plot



Solution Line: multiples of (-5, 1, -2)
6
4
2
0
−2
−4
−6
Z
3
2
1
0
−1
−2
−3
Y
−10
−5
0
5
10
X

Figure

Code - C

```
#include <stdio.h>

#define ROWS 3
#define COLS 4

// Simple row reduction for 3x4 augmented matrix
void row_reduce(double A[ROWS][COLS]) {
    // Eliminate below pivot (0,0)
    if (A[0][0] != 0) {
        for (int i = 1; i < ROWS; i++) {
            double factor = A[i][0] / A[0][0];
            for (int j = 0; j < COLS; j++) {
                A[i][j] -= factor * A[0][j];
            }
        }
    }
```

Code - C

```
    // Eliminate below pivot (1,1)
    if (A[1][1] != 0) {
        for (int i = 2; i < ROWS; i++) {
            double factor = A[i][1] / A[1][1];
            for (int j = 0; j < COLS; j++) {
                A[i][j] -= factor * A[1][j];
            }
        }
    }
}
```

Code - Python(with shared C code)

The code to obtain the required plot is

```
import ctypes
import numpy as np
import matplotlib.pyplot as plt

# --- Load shared library ---
lib = ctypes.CDLL("./solver.so")

# Define ctypes type
Mat3x4 = (ctypes.c_double * 4) * 3
lib.row_reduce.argtypes = [Mat3x4]
lib.row_reduce.restype = None

# --- Step 1: Build augmented matrix with variable k ---
k = 11 # try different k values here
A4 = Mat3x4()
```

Code - Python(with shared C code)

```
A4[0][:] = (1, k, 3, 0)
A4[1][:] = (3, k, -2, 0)
A4[2][:] = (2, 4, -3, 0)

print("Original augmented matrix:")
for row in A4:
    print([float(x) for x in row])

# --- Step 2: Row-reduce using C ---
lib.row_reduce(A4)

print("\nRow-reduced augmented matrix:")
reduced = np.array([[A4[i][j] for j in range(4)] for i in range(3)], dtype=
    float)
print(reduced)
```

Code - Python(with shared C code)

```
# --- Step 3: Check rank (ignoring last column) ---
coeff_matrix = reduced[:, :3]
rank = np.linalg.matrix_rank(coeff_matrix)
print("\nRank of coefficient matrix =", rank)

if rank == 3:
    print("Only trivial solution.")
    exit()

# --- Step 4: Solve system (from reduced form) ---
# From reduced system when k=11:
# Row2: 2y + z = 0 -> z = -2y
# Row1: x + 11y + 3z = 0 -> x = -5y
solution_vec = np.array([-5.0, 1.0, -2.0])
print("Solution vector:", solution_vec)
```

Code - Python(with shared C code)

```
# Ratio
x, y, z = solution_vec
ratio = (x * z) / (y * y)
print("xz / y^2 =", ratio)

# --- Step 5: Plot solution line ---
t_vals = np.linspace(-2, 2, 200)
points = t_vals[:, None] * solution_vec[None, :]

fig = plt.figure()
ax = fig.add_subplot(111, projection="3d")
ax.plot(points[:, 0], points[:, 1], points[:, 2], "b-", linewidth=2)
ax.set_xlabel("X"); ax.set_ylabel("Y"); ax.set_zlabel("Z")
ax.set_xlim([-12, 12]); ax.set_ylim([-3, 3]); ax.set_zlim([-6, 6])
ax.set_title("Solution Line: multiples of (-5, 1, -2)")
plt.savefig("solvek.png")
plt.show()
```

Code - Python only

```
import numpy as np
import matplotlib.pyplot as plt

# ----- Set k(solving with k as variable requires some concepts like
    nullspace which aren't taught yet) -----
k = 11.0 # change this to any real value

# ----- Build augmented matrix -----
A = np.array([
    [1, k, 3, 0],
    [3, k, -2, 0],
    [2, 4, -3, 0]
], dtype=float)

print("Original augmented matrix:")
print(A, "\n")
```

Code - Python only

```
# ---- Row reduction ----
# Step 1: eliminate below pivot (0,0)
if A[0,0] != 0:
    A[1] = A[1] - (A[1,0]/A[0,0])*A[0]
    A[2] = A[2] - (A[2,0]/A[0,0])*A[0]

# Step 2: eliminate below pivot (1,1)
if A[1,1] != 0:
    A[2] = A[2] - (A[2,1]/A[1,1])*A[1]

print("Row reduced matrix:")
print(A, "\n")

# ---- Check rank ----
rank = np.linalg.matrix_rank(A[:,:3])
print("rank =", rank)
```

Code - Python only

```
if rank == 3:
    print("Only trivial solution.")
else:
    # For k=11, reduced system gives:
    # 2y + z = 0 -> z = -2y
    # x + 11y + 3z = 0 -> x = -5y
    y = 1
    x = -5*y
    z = -2*y
    solution_vec = np.array([x,y,z], dtype=float)
    print("Solution vector:", solution_vec)

    ratio = (x*z)/(y*y)
    print("xz / y^2 =", ratio)
```

Code - Python only

```
# ---- Plot the solution line ----
t = np.linspace(-2, 2, 200)
line = np.outer(t, solution_vec)

fig = plt.figure()
ax = fig.add_subplot(111, projection="3d")
ax.plot(line[:,0], line[:,1], line[:,2], 'b-')
ax.set_xlabel("X"); ax.set_ylabel("Y"); ax.set_zlabel("Z")
ax.set_title(f"Solution line for k={k}")
plt.savefig("newsolvek.png")
plt.show()
```