5.9.1

RAVULA SHASHANK REDDY - EE25BTECH11047

October 1, 2025

Question

A fraction becomes $\frac{1}{3}$ when 1 is subtracted from its numerator and it becomes $\frac{1}{4}$ when 8 is added to its denominator. Find the fraction.

Solution

Let fraction be $\frac{x}{y}$

$$\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix} \tag{1}$$

$$\frac{x-1}{y} = \frac{1}{3} \tag{2}$$

$$\begin{pmatrix} 3 \\ -1 \end{pmatrix}^T \mathbf{x} = 3 \tag{3}$$

$$\frac{x}{y+8} = \frac{1}{4} \tag{4}$$

$$\begin{pmatrix} 4 \\ -1 \end{pmatrix}^T \mathbf{x} = 8 \tag{5}$$

$$\begin{pmatrix} 3 & -1 \\ 4 & -1 \end{pmatrix} \mathbf{x} = \begin{pmatrix} 3 \\ 8 \end{pmatrix} \tag{6}$$

Solution

$$\begin{pmatrix} 3 & -1 & 3 \\ 4 & -1 & 8 \end{pmatrix} \xrightarrow{R_2 \to R_2 - R_1} \begin{pmatrix} 3 & -1 & 3 \\ 1 & 0 & 5 \end{pmatrix} \xrightarrow{R_1 \to R_1 - 3R_2} \begin{pmatrix} 0 & -1 & -12 \\ 1 & 0 & 5 \end{pmatrix} \quad (7)$$

$$\mathbf{x} = \begin{pmatrix} 5 \\ 12 \end{pmatrix} \quad (8)$$

$$\boxed{Required \quad Fraction = \frac{5}{12}}$$

C Code

```c
#include <stdio.h>

int main() {
    // Augmented matrix [A|b]
    double mat[2][3] = {
        {3, -1, 3}, // 3x - y = 3
        {4, -1, 8} // 4x - y = 8
    };

    // Row operation: R2 -> R2 - (4/3)R1
    double factor = mat[1][0] / mat[0][0];
    for (int j = 0; j < 3; j++) {
        mat[1][j] = mat[1][j] - factor * mat[0][j];
    }
```

C Code

```c
    // Scale R2 to make pivot = 1
    double scale = mat[1][1];
    for (int j = 0; j < 3; j++) {
        mat[1][j] /= scale;
    }

    // Back substitution
    double y = mat[1][2];
    double x = (mat[0][2] - mat[0][1] * y) / mat[0][0];

    printf(x = %.2f\n, x);
    printf(y = %.2f\n, y);
    printf(Fraction = %.2f/%.2f\n, x, y);

    return 0;
}
```

Python Direct

```python
    import numpy as np
import libs.line.funcs as line
import libs.triangle.funcs as tri

# Coefficient matrix A and RHS vector b
A = np.array([[3, -1],
              [4, -1]], dtype=float)
b = np.array([3, 8], dtype=float)

# Solve system A x = b
x = np.linalg.solve(A, b)

num, den = x[0], x[1]

print(x =, num)
print(y =, den)
print(Fraction = {}/{}.format(int(num), int(den)))
```

Python Shared

```
    import numpy as np
import ctypes

# Load shared object
solver = ctypes.CDLL(./solver.so)

# Define return types and arguments
solver.solve.argtypes = [np.ctypeslib.ndpointer(dtype=np.float64,
    ndim=1, flags=C)]

# Prepare output array
out = np.zeros(2, dtype=np.float64)
```

Python Shared

```python
# Call C function
solver.solve(out)

x, y = out[0], out[1]

print(x =, x)
print(y =, y)
print(Fraction = {}/{}.format(int(x), int(y)))
```