

5.3.29

AI25BTECH11035-SUJAL RAJANI

September 5,2025

Question

Solve for the system of linear equations:

$$x + 2y = 6$$

$$2x - 5y = 12$$

Theoretical Solution

According to the question,
The equation of lines given,

$$\begin{pmatrix} 1 & 2 \end{pmatrix} \mathbf{x} = 6 \quad \begin{pmatrix} 2 & -5 \end{pmatrix} \mathbf{x} = 12 \quad (1)$$

$$\therefore \begin{pmatrix} 1 & 2 \\ 2 & -5 \end{pmatrix} \mathbf{x} = \begin{pmatrix} 6 \\ 12 \end{pmatrix} \quad (2)$$

Theoretical Solution

Forming an augmented matrix,

$$\left(\begin{array}{cc|c} 1 & 2 & 6 \\ 2 & -5 & 12 \end{array} \right) \quad (3)$$

Upon doing row reduction,

$$\left(\begin{array}{cc|c} 1 & 2 & 6 \\ 2 & -5 & 12 \end{array} \right) \xleftrightarrow{R_2 \leftarrow R_2 - 2 \times R_1} \left(\begin{array}{cc|c} 1 & 2 & 6 \\ 0 & -9 & 0 \end{array} \right) \xleftrightarrow{R_1 \leftarrow R_1 + \frac{2}{9} \times R_2} \left(\begin{array}{cc|c} 1 & 0 & 6 \\ 0 & -9 & 0 \end{array} \right) \quad (4)$$

$$\Rightarrow \mathbf{x} = \begin{pmatrix} 6 \\ 0 \end{pmatrix} \quad (5)$$

`../figs/Figure.png`

C Code -Finding Solution for the system of Equations

```
#include <stdio.h>

void rref_solver(double aug[2][3], double solution[2]) {
    // Normalize first row (pivot = aug[0][0])
    double pivot = aug[0][0];
    for (int j = 0; j < 3; j++) {
        aug[0][j] /= pivot;
    }
    // Eliminate below pivot
    double factor = aug[1][0];
    for (int j = 0; j < 3; j++) {
        aug[1][j] -= factor * aug[0][j];
    }
}
```

C Code -Finding Solution for the system of Equations

```
// Normalize second row (pivot = aug[1][1])
pivot = aug[1][1];
for (int j = 0; j < 3; j++) {
    aug[1][j] /= pivot;
}
// Eliminate above pivot
factor = aug[0][1];
for (int j = 0; j < 3; j++) {
    aug[0][j] -= factor * aug[1][j];
}
// Extract solution
solution[0] = aug[0][2]; // x
solution[1] = aug[1][2]; // y
}
```

```
import ctypes
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mp
mp.use("TkAgg")
# Load the shared C library
lib = ctypes.CDLL("./liblineq_solver.so")
# Define argument and return types
lib.rref_solver.argtypes = [ctypes.c_double * 6, ctypes.c_double * 2]
# Create augmented matrix for system:
aug = (ctypes.c_double * 6)(1, 2, 6, 2, -5, 12) # Flattened 2x3
solution = (ctypes.c_double * 2)()
lib.rref_solver(aug, solution)
# Convert result to numpy vector (ensure flat)
x_sol = np.array([solution[0], solution[1]], dtype=float).flatten()
print("Solution vector from C:", x_sol)
```



```
# plot
x_vals = np.linspace(-2, 10, 400)
y1 = (6 - x_vals) / 2
y2 = (12 - 2*x_vals) / -5

plt.plot(x_vals, y1, label=r"$x+3y=6$")
plt.plot(x_vals, y2, label=r"$2x-3y=12$")

plt.scatter(x_sol[0], x_sol[1], color="red", zorder=5)
plt.text(float(x_sol[0])+0.2, float(x_sol[1]), f"({x_sol[0]:.1f},
{x_sol[1]:.1f})", color="red")
```

```
plt.xlabel("x")
plt.ylabel("y")
plt.title("Garphical solution of the Linear system")
plt.axhline(0, color="black", linewidth=0.8)
plt.axvline(0, color="black", linewidth=0.8)
plt.legend()
plt.grid(True)
plt.show()
```

```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mp
mp.use("TkAgg")

A=np.array([[1,2],[2,-5]],dtype=float)
b=np.array([6,12], dtype=float)

x=np.linalg.solve(A,b)
print("Solution vector for the system of equations:",x)
```

```
# Making a plot
```

```
x_vals = np.linspace(-2, 10, 400)
```

```
# Rearranged equations to express y in terms of x
```

```
y1 = (6 - x_vals) / 2 # from  $x + 3y = 6$ 
```

```
y2 = (12 - 2*x_vals) / -5 # from  $2x - 3y = 12$ 
```

```
# Plot lines
```

```
plt.plot(x_vals, y1, label=r"$x + 2y = 6$")
```

```
plt.plot(x_vals, y2, label=r"$2x - 5y = 12$")
```

```
# Mark solution
```

```
plt.scatter(x[0], x[1], color="red", zorder=5)
```

```
plt.text(x[0]+0.2, x[1], f"({x[0]:.1f}, {x[1]:.1f})", color="red")
```

```
# Formatting
plt.xlabel("x")
plt.ylabel("y")
plt.title("Graphical Solution of the Linear System")
plt.axhline(0, color='black', linewidth=0.8)
plt.axvline(0, color='black', linewidth=0.8)
plt.legend()
plt.grid(True)
plt.show()
```

`../figs/Figure_1.png`