6.2.6

Namaswi-EE25BTECH11060

september 2025

Question

Find matrix X such that

$$X \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} = \begin{pmatrix} -7 & -8 & -9 \\ 2 & 4 & 6 \end{pmatrix} \tag{1}$$

Solution

Form the augmented matrix

$$\begin{pmatrix} 1 & 4 & | & -7 & -8 & -9 \\ 2 & 5 & | & 2 & 4 & 6 \\ 3 & 6 & | & 0 & 0 & 0 \end{pmatrix} \tag{2}$$

$$\tag{3}$$

Replace $R_2 \to R_2 - 2R_1$ and $R_3 \to R_3 - 3R_1$

$$\begin{pmatrix} 1 & 4 & | & -7 & -8 & -9 \\ 0 & -3 & | & 16 & 20 & 27 \\ 0 & -6 & | & 21 & 24 & 27 \end{pmatrix} \tag{4}$$

Solution

Replace $R_2 \rightarrow \frac{-1}{3} R_2$ and $R_3 \rightarrow R_3 - 2R_2$

$$\begin{pmatrix} 1 & 4 & | & -7 & -8 & -9 \\ 0 & 1 & | & -16/3 & -20/3 & -9 \\ 0 & 0 & | & -11/3 & -16/3 & 9 \end{pmatrix} \tag{5}$$

Hence,

$$\mathbf{X} = \begin{pmatrix} 1 & 2 \\ -2 & 0 \end{pmatrix} \tag{6}$$

Solution

Pseudoinverse verification

Let,

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \tag{7}$$

$$\mathbf{B} = \begin{pmatrix} -7 & 2 \\ -8 & 4 \\ -9 & 6 \end{pmatrix} \tag{8}$$

$$\mathbf{A}^{+} = \mathbf{A}^{\top}(\mathbf{A}\mathbf{A}^{\top})^{-1} \tag{9}$$

$$= \begin{pmatrix} -17/18 & 4/9 \\ -1/9 & 1/9 \\ 13/18 & -2/9 \end{pmatrix} \tag{10}$$

Solution

$$\mathbf{X} = \mathbf{B}\mathbf{A}^{+} \tag{11}$$

$$= \begin{pmatrix} -7 & 2 \\ -8 & 4 \\ -9 & 6 \end{pmatrix} \begin{pmatrix} -17/18 & 4/9 \\ -1/9 & 1/9 \\ 13/18 & -2/9 \end{pmatrix} \tag{12}$$

$$= \begin{pmatrix} 1 & 2 \\ -2 & 0 \end{pmatrix} \tag{13}$$

Python Code

```
    import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Create a grid of a and b values
a = np.linspace(-10, 10, 20)
b = np.linspace(-10, 10, 20)
A, B = np.meshgrid(a, b)

# Define the planes
Z1 = -7 - 1*A - 4*B # a + 4b + z = -7 => z = -7 - a - 4b
Z2 = -8 - 2*A - 5*B # 2a + 5b + z = -8 => z = -8 - 2a - 5b
Z3 = -9 - 3*A - 6*B # 3a + 6b + z = -9 => z = -9 - 3a - 6b
```

Python Code

```
    # Plot
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(A, B, Z1, alpha=0.5, color='red', label='Plane 1')
ax.plot_surface(A, B, Z2, alpha=0.5, color='green', label='Plane 2')
ax.plot_surface(A, B, Z3, alpha=0.5, color='blue', label='Plane 3')

ax.set_xlabel('a')
ax.set_ylabel('b')
ax.set_zlabel('c')
ax.set_title('Graph of 3 Planes')

plt.show()
```

C Code

```c
    #include <stdio.h>
int main() {
    int i, j, k;
    double a[3][3] = {
        {1, 4, -7}, // a + 4b = -7
        {2, 5, -8}, // 2a + 5b = -8
        {3, 6, -9} // 3a + 6b = -9
    };
    double factor;
```

C Code

```
// Forward elimination
for (i = 0; i < 2; i++) { // only first 2 rows, since 2
    variables
    for (j = i+1; j < 3; j++) {
        if(a[i][i] != 0){
            factor = a[j][i] / a[i][i];
            for (k = i; k < 3; k++) {
                a[j][k] -= factor * a[i][k];
            }
        }
    }
}
```

C Code

```
    // Back substitution
    double b_val, a_val;
    if(a[1][1] != 0){
        b_val = a[1][2] / a[1][1];
        a_val = (a[0][2] - 4*b_val) / 1;
        printf("Solution: a = %.2lf, b = %.2lf\n", a_val, b_val);
    } else {
        printf("No unique solution exists.\n");
    }
    return 0;
}
```

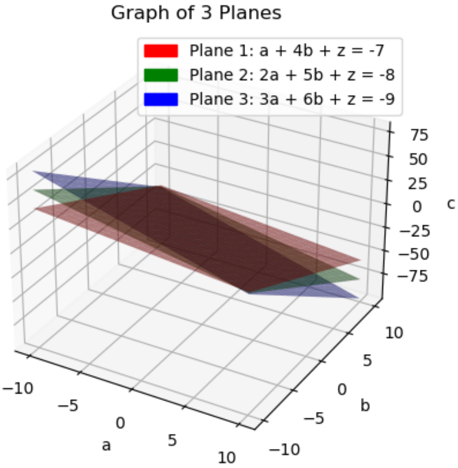
C and Python Code

```
   import ctypes

# Load shared library
lib = ctypes.CDLL('./libsolver.so')

# Prepare variables
a = ctypes.c_double()
b = ctypes.c_double()
status = ctypes.c_int()
```

C and Python Code

```python
    # Call C function
lib.solve_system(ctypes.byref(a), ctypes.byref(b), ctypes.byref(
    status))

# Check result
if status.value == 1:
    print(f"Solution from C: a = {a.value}, b = {b.value}")
else:
    print("No unique solution exists.")
```

Plot


Graph of 3 Planes
Plane 1: a + 4b + z = -7
Plane 2: 2a + 5b + z = -8
Plane 3: 3a + 6b + z = -9
75
50
25
0
−25
−50
−75
c
10
5
0
−5
−10
b
−10
−5
0
5
10
a