Assignment

Determinant of Matrix M:

Matrix M1:

$$M_1 = \begin{pmatrix} 17 & -11 \\ 6 & -3 \end{pmatrix}$$

The determinant of a 2x2 matrix is calculated as:

det(M1)= (a.d)-(b.c)

For matrix M_1:

det(M1) (17.-3)-(-11.6)=- 51+66 =15

Matrix M2:

$$M_2 = \begin{pmatrix} 1 & 1 & 2 \\ 2 & 3 & 1 \\ 3 & 4 & -5 \end{pmatrix}$$

For a 3x3 matrix, the determinant is calculated as:

det(M) = a(ei-fh)-b(di-fg) + c(dh - eg)

For matrix M2:

det(M2)=1((3.-5)- (1.4))-1((2.-5)- (1.3))+2((2.4)- (3.3))

Simplifying:

det(M2)=1(-15-4)1(-10-3)+2(8-9)

det(M2)=-19-13-2=-8

The code part:

```python
# 1. Determinant of M1 and M2
M1 = np.array([[17, -11], [6, -3]])
det_M1 = np.linalg.det(M1)

M2 = np.array([[1, 1, 2], [2, 3, 1], [3, 4, -5]])
det_M2 = np.linalg.det(M2)

print("Determinant of M1:", det_M1)
print("Determinant of M2:", det_M2)
```

```
Determinant of M1: 14.999999999999993
Determinant of M2: -7.999999999999998
```

2. **Inverse of Matrix A:**

Matrix A1:

$$A_1 = \begin{pmatrix} -3 & -2 \\ 3 & 3 \end{pmatrix}$$

The inverse of a 2x2 matrix A^-1 is given by:

$$A^{-1} = \frac{1}{\det(A)} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

Where the determinant of A1 is:

det(A1) = (-3) (3)-(-2)(3) = -9+6=-3

So the inverse is:

$$A_1^{-1} = \frac{1}{-3}\begin{pmatrix} 3 & 2 \\ -3 & -3 \end{pmatrix} = \begin{pmatrix} -1 & -\frac{2}{3} \\ 1 & 1 \end{pmatrix}$$

Matrix A2:

$$A_2 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

THE code part:

```
[3] # 2. Inverse of A1 and A2
    A1 = np.array([[-3, -2], [3, 3]])
    inv_A1 = np.linalg.inv(A1)

    A2 = np.array([[1, 0, 1], [0, 1, 1], [1, 1, 1]])
    inv_A2 = np.linalg.inv(A2)

    print("Inverse of A1:")
    print(inv_A1)

    print("Inverse of A2:")
    print(inv_A2)
```

```
Inverse of A1:
[[-1.         -0.66666667]
 [ 1.          1.        ]]
Inverse of A2:
[[ 0. -1.  1.]
 [-1.  0.  1.]
 [ 1.  1. -1.]]
```

3. **Solution Set for Systems of Linear Equations:**

Problem a)

Given system of equations:

$$\begin{aligned} &1. \quad x_1 + 4x_2 + 3x_3 - x_4 = 5 \\ &2. \quad x_1 - x_2 + x_3 + 2x_4 = 6 \\ &3. \quad 4x_1 + x_2 + 6x_3 + 5x_4 = 9 \end{aligned}$$

Step 1: Write the augmented matrix:

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

Step 2: Perform Gaussian elimination

Row Operation 1: Subtract Row 1 from Row 2:

$$R2 = R2 - R1 \quad \Rightarrow \quad \begin{pmatrix} 1 & 4 & 3 & -1 & | & 5 \\ 0 & -5 & -2 & 3 & | & 1 \\ 4 & 1 & 6 & 5 & | & 9 \end{pmatrix}$$

Row Operation 2: Subtract 4 times Row 1 from Row 3:

$$R3 = R3 - 4R1 \quad \Rightarrow \quad \begin{pmatrix} 1 & 4 & 3 & -1 & | & 5 \\ 0 & -5 & -2 & 3 & | & 1 \\ 0 & -15 & -6 & 9 & | & -11 \end{pmatrix}$$

Row Operation 3: Multiply Row 2 by −1/5 to normalize the leading coefficient:

$$R2 = -\frac{1}{5}R2 \quad \Rightarrow \quad \begin{pmatrix} 1 & 4 & 3 & -1 & | & 5 \\ 0 & 1 & \frac{2}{5} & -\frac{3}{5} & | & -\frac{1}{5} \\ 0 & -15 & -6 & 9 & | & -11 \end{pmatrix}$$

Row Operation 4: Add 15 times Row 2 to Row 3 to eliminate x2 from Row 3:

$$R3 = R3 + 15R2 \quad \Rightarrow \quad \left(\begin{array}{cccc|c} 1 & 4 & 3 & -1 & 5 \\ 0 & 1 & \frac{2}{5} & -\frac{3}{5} & -\frac{1}{5} \\ 0 & 0 & 0 & 0 & -14 \end{array}\right)$$

Problem b:

Given system of equations:

$$\begin{aligned} &1. \quad x_1 - 2x_2 + x_3 - x_4 = 3 \\ &2. \quad 2x_1 - 4x_2 + x_3 + x_4 = 2 \\ &3. \quad x_1 - 2x_2 - 2x_3 + 3x_4 = 1 \end{aligned}$$

Step 1: Write the augmented matrix :

$$\left(\begin{array}{cccc|c} 1 & -2 & 1 & -1 & 3 \\ 2 & -4 & 1 & 1 & 2 \\ 1 & -2 & -2 & 3 & 1 \end{array}\right)$$

Step 2: Perform Gaussian elimination

- **Row Operation 1:** Subtract 2 times Row 1 from Row 2:

$$R2 = R2 - 2R1 \quad \Rightarrow \quad \left(\begin{array}{cccc|c} 1 & -2 & 1 & -1 & 3 \\ 0 & 0 & -1 & 3 & -4 \\ 1 & -2 & -2 & 3 & 1 \end{array}\right)$$

- **Row Operation 2:** Subtract Row 1 from Row 3:

$$R3 = R3 - R1 \quad \Rightarrow \quad \left(\begin{array}{cccc|c} 1 & -2 & 1 & -1 & 3 \\ 0 & 0 & -1 & 3 & -4 \\ 0 & 0 & -3 & 4 & -2 \end{array}\right)$$

- **Row Operation 3:** Add 3 times Row 2 to Row 3:

$$R3 = R3 + 3R2 \quad \Rightarrow \quad \left(\begin{array}{cccc|c} 1 & -2 & 1 & -1 & 3 \\ 0 & 0 & -1 & 3 & -4 \\ 0 & 0 & 0 & 13 & -14 \end{array}\right)$$

Step 3: Back-substitution:

- From the last row:

$$13x_4 = -14 \quad \Rightarrow \quad x_4 = -\frac{14}{13}$$

- From the second row:

$$-x_3 + 3\left(-\frac{14}{13}\right) = -4 \quad \Rightarrow \quad x_3 = -\frac{10}{13}$$

- From the first row:

$$x_1 - 2x_2 + \left(-\frac{10}{13}\right) - \left(-\frac{14}{13}\right) = 3 \quad \Rightarrow \quad x_1 - 2x_2 + \frac{4}{13} = 3$$

This equation shows x1 as a function of x2, indicating **infinitely many solutions**.

Conclusion:

The system has infinitely many solutions, with x expressed as a function of x2

Problem c)

Given system of equations:

$$\begin{aligned} &1. \quad x_1 + 2x_2 + 3x_3 = 1 \\ &2. \quad 2x_1 - x_2 + x_3 = 2 \\ &3. \quad 3x_1 + x_2 + x_3 = 4 \end{aligned}$$

Step 1: Write the augmented matrix:

$$\begin{pmatrix} 1 & 2 & 3 & | & 1 \\ 2 & -1 & 1 & | & 2 \\ 3 & 1 & 1 & | & 4 \end{pmatrix}$$

Step 2: Perform Gaussian elimination

- **Row Operation 1:** Subtract 2 times Row 1 from Row 2:

$$R2 = R2 - 2R1 \quad \Rightarrow \quad \begin{pmatrix} 1 & 2 & 3 & | & 1 \\ 0 & -5 & -5 & | & 0 \\ 3 & 1 & 1 & | & 4 \end{pmatrix}$$

- **Row Operation 2:** Subtract 3 times Row 1 from Row 3:

$$R3 = R3 - 3R1 \quad \Rightarrow \quad \begin{pmatrix} 1 & 2 & 3 & | & 1 \\ 0 & -5 & -5 & | & 0 \\ 0 & -5 & -8 & | & 1 \end{pmatrix}$$

- **Row Operation 3:** Add Row 2 to Row 3:

$$R3 = R3 + R2 \quad \Rightarrow \quad \begin{pmatrix} 1 & 2 & 3 & | & 1 \\ 0 & -5 & -5 & | & 0 \\ 0 & 0 & -13 & | & 1 \end{pmatrix}$$

Step 3: Back-substitution:

- From the last row:

$$-13x_3 = 1 \quad \Rightarrow \quad x_3 = -\frac{1}{13}$$

- From the second row:

$$-5x_2 - 5\left(-\frac{1}{13}\right) = 0 \quad \Rightarrow \quad x_2 = \frac{1}{13}$$

Code part:

```
# Coefficient matrix A and vector b for Problem a
A_a = np.array([
    [1, 4, 3, -1],
    [1, -1, 1, 2],
    [4, 1, 6, 5]
])
b_a = np.array([5, 6, 9])

x_a = solve_least_squares(A_a, b_a)
print("Least squares solution for system a:", x_a)
```

```
Least squares solution for system a: [0.52334152 0.2997543  0.85257985 0.55282555]
```

```
# Coefficient matrix A and vector b for Problem c
A_c = np.array([
    [1, 1, 2, -1],
    [3, 5, 8, -4],
    [0, 1, 0, 1]
])
b_c = np.array([2, 3, 1])

x_b = solve_least_squares(A_b, b_b)
print("Least squares solution for system b:", x_b)
```

```
Least squares solution for system b: [ 0.6 -1.2 -2.  -2. ]
```

```
# Coefficient matrix A and vector b for Problem c
A_c = np.array([
    [1, 1, 2, -1],
    [3, 5, 8, -4],
    [0, 1, 0, 1]
])
b_c = np.array([2, 3, 1])

x_c = solve_least_squares(A_c, b_c)
print("Least squares solution for system c:", x_c)
```

```
Least squares solution for system c: [ 3.95238095 -1.04761905  0.57142857  2.04761905]
```