[1.1] ex. 9(a), (b)

Part (a): Show that the system will have a unique solution if $m_1 \neq m_2$

To prove this, we can use the determinant of the coefficient matrix of the system. The system can be written in matrix form as:

$$Ax = b$$

$$\begin{bmatrix} -m_1 & 1 \\ -m_2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

The determinant of a matrix A is:

$$\det(A) = (-m_1) \cdot 1 - (-m_2) \cdot 1 = -m_1 + m_2$$

For the system to have a unique solution, the determinant of the coefficient matrix A must not be zero:

This condition guarantees that the system of equations has a unique solution, as it ensures that A is invertible.

$$-m_1 + m_2 \neq 0 \implies m_1 \neq m_2$$

Part (b): Show that if $m_1 = m_2$, then the system will be consistent only if $b_1 = b_2$.

If $m_1 = m_2$, then the 2 equations become essentially the same equation multiplied by a constant, provided $b_1 = b_2$. In this case, the determinant of A is zero $(m_1 - m_2 = 0)$, indicating that the system may either have no solution or infinitely many solutions (the system is degenerate).

For the system to be consistent (i.e., at least one solution exists), the second equation must not introduce any contradiction to the first one. This happens if and only if $b_1 = b_2$ when $m_1 = m_2$. Essentially, both equations describe the same line in this scenario, leading to infinitely many solutions lying on this line.

Editor - D:\Linear Algebra\D84099084_HW1\HW1_1_1_9ab.m

HW1_1_1_9ab.m | HW1_1_1_9c.m | +

```matlab
function HW1_1_1_9ab(m1, m2, b1, b2)
    % Coefficient matrix A
    A = [-m1 1; -m2 1];

    % Right-hand side vector b
    b = [b1; b2];

    % Calculate the determinant of A
    detA = det(A);

    % Check for the uniqueness of the solution (m1 != m2)
    if detA ~= 0
        fprintf('The system has a unique solution since m1 != m2 (det(A) = %f).\n', detA);
    else
        % If m1 = m2, check for consistency (b1 = b2)
        if b1 == b2
            fprintf('The system is consistent (has infinitely many solutions) since m1 = m2 and b1 = b2.\n');
        else
            fprintf('The system is inconsistent (no solution) since m1 = m2 but b1 != b2.\n');
        end
    end
end

```

Command Window

```
>> HW1_1_1_9ab(1, 2, 3, 4); % Checks for a unique solution with m1 ≠ m2
HW1_1_1_9ab(1, 1, 3, 3); % Checks for consistency with m1 = m2 and b1 = b2
HW1_1_1_9ab(1, 1, 3, 4); % Checks for inconsistency with m1 = m2 but b1 ≠ b2
The system has a unique solution since m1 != m2 (det(A) = 1.000000).
The system is consistent (has infinitely many solutions) since m1 = m2 and b1 = b2.
The system is inconsistent (no solution) since m1 = m2 but b1 != b2.
```

賴溡雨 D84099084_HW1

Shih-Yu, Lai

[1.2] ex. 6

System (a)

Equations:

$$2x + y = 1$$
$$7x + 6y = 1$$

Augmented Matrix:

$$\left[\begin{array}{cc|c} 2 & 1 & 1 \\ 7 & 6 & 1 \end{array}\right]$$

Steps:

- Multiply Row 1 by 1/2 to make the leading coefficient 11.

$$\left[\begin{array}{cc|c} 1 & \frac{1}{2} & \frac{1}{2} \\ 7 & 6 & 1 \end{array}\right]$$

- Subtract 7×Row 1 from Row 2.

$$\left[\begin{array}{cc|c} 1 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{5}{2} & -\frac{5}{2} \end{array}\right]$$

- Multiply Row 2 by 2/5.

$$\left[\begin{array}{cc|c} 1 & \frac{1}{2} & \frac{1}{2} \\ 0 & 1 & -1 \end{array}\right]$$

- Subtract 1/2× Row 2 from Row 1.

$$\left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & -1 \end{array}\right]$$

Solution:

$$x = 1, y = -1$$

System (b)

Equations:

$$x_1 + x_2 - x_3 + x_4 = 6$$
$$2x_1 - x_2 + x_3 - x_4 = -3$$
$$3x_1 + x_2 - 2x_3 + x_4 = 9$$

Augmented Matrix:

$$\left[\begin{array}{cccc|c} 1 & 1 & -1 & 1 & 6 \\ 2 & -1 & 1 & -1 & -3 \\ 3 & 1 & -2 & 1 & 9 \end{array}\right]$$

Steps:

- Multiply Row 2 by 1/2 and subtract Row 1 from it.

$$\left[\begin{array}{cccc|c} 1 & 1 & -1 & 1 & 6 \\ 0 & -\frac{3}{2} & \frac{3}{2} & -\frac{3}{2} & -6 \\ 3 & 1 & -2 & 1 & 9 \end{array}\right]$$

- Multiply Row 3 by 1/3 and subtract 3 times Row 1 from it.

$$\left[\begin{array}{cccc|c} 1 & 1 & -1 & 1 & 6 \\ 0 & -\frac{3}{2} & \frac{3}{2} & -\frac{3}{2} & -6 \\ 0 & -2 & 1 & -2 & -9 \end{array}\right]$$

- Multiply Row 2 by −2/3 and add it to Row 3.

$$\left[\begin{array}{cccc|c} 1 & 1 & -1 & 1 & 6 \\ 0 & 1 & -1 & 1 & 4 \\ 0 & 0 & -1 & -1 & -7 \end{array}\right]$$

- Divide Row 3 by −1.

$$\left[\begin{array}{cccc|c} 1 & 1 & -1 & 1 & 6 \\ 0 & 1 & -1 & 1 & 4 \\ 0 & 0 & 1 & 1 & 7 \end{array}\right]$$

- Add Row 3 to Row 2 and Row 1.

$$\left[\begin{array}{cccc|c} 1 & 1 & 0 & 2 & 13 \\ 0 & 1 & 0 & 2 & 11 \\ 0 & 0 & 1 & 1 & 7 \end{array}\right]$$

- Subtract Row 2 from Row 1.

$$\left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 2 & 11 \\ 0 & 0 & 1 & 1 & 7 \end{array}\right]$$

Solution:

$$x_1 = 2, x_2 = -1, x_3 = 3, x_4 = 4$$

System (c)

Equations:

$$x_1 - 10x_2 + 5x_3 = -4$$
$$x_1 + x_2 + x_3 = 1$$

Augmented Matrix:

$$\left[\begin{array}{ccc|c} 1 & -10 & 5 & -4 \\ 1 & 1 & 1 & 1 \end{array}\right]$$

Steps:

- Subtract Row 1 from Row 2 to get a leading zero in the second row, first column.

$$\left[\begin{array}{ccc|c} 1 & -10 & 5 & -4 \\ 0 & 11 & -4 & 5 \end{array}\right]$$

- Divide Row 2 by 11.

$$\left[\begin{array}{ccc|c} 1 & -10 & 5 & -4 \\ 0 & 1 & -\frac{4}{11} & \frac{5}{11} \end{array}\right]$$

- Add 10 times Row 2 to Row 1.

$$\left[\begin{array}{ccc|c} 1 & 0 & \frac{10}{11} & \frac{2}{11} \\ 0 & 1 & -\frac{4}{11} & \frac{5}{11} \end{array}\right]$$

- Multiply Row 1 by 11 and subtract 10 times Row 2 from Row 1 to get a leading one.

$$\left[\begin{array}{ccc|c} 11 & 0 & 10 & 2 \\ 0 & 1 & -\frac{4}{11} & \frac{5}{11} \end{array}\right]$$

- Finally, divide Row 1 by 11 to get a leading one.

$$\left[\begin{array}{ccc|c} 1 & 0 & \frac{10}{11} & \frac{2}{11} \\ 0 & 1 & -\frac{4}{11} & \frac{5}{11} \end{array}\right]$$

Solution:

$$x_1 = \frac{2}{11}, x_2 = \frac{5}{11}, x_3 = 0$$

Editor - D:\Linear Algebra\D84099084_HW1\HW1_1_2_6.m

HW1_1_1_9ab.m | HW1_1_1_9c.m | HW1_1_Matlab_ex_1.m | HW1_1_2_15.m

```matlab
% System (a)
A_a = [2 1; 7 6];
b_a = [1; 1];
x_a = gauss_jordan_elimination(A_a, b_a);

% System (b)
A_b = [1 1 -1 1; 2 -1 1 -1; 3 1 -2 1];
b_b = [6; -3; 9];
x_b = gauss_jordan_elimination(A_b, b_b);

% System (c)
A_c = [1 -10 5; 1 1 1];
b_c = [-4; 1];
x_c = gauss_jordan_elimination(A_c, b_c);

% System (d)
A_d = [1 -2 3 1; 2 1 -1 1; 1 3 1 1];
b_d = [4; 1; 3];
x_d = gauss_jordan_elimination(A_d, b_d);

% Display the results
disp('Solution for system (a):');
disp(x_a);

disp('Solution for system (b):');
disp(x_b);

disp('Solution for system (c):');
disp(x_c);

disp('Solution for system (d):');
disp(x_d);

% Gauss-Jordan elimination
function x = gauss_jordan_elimination(A, b)
    % Form the augmented matrix
    aug = [A b];

    % Number of equations
    n = size(aug, 1);

    % Forward elimination to get upper triangular matrix
    for i = 1:n
        % Make the diagonal element 1
        aug(i,:) = aug(i,:) / aug(i,i);

        % Make the off-diagonal elements in column i zero
        for j = 1:n
            if i ~= j
                aug(j,:) = aug(j,:) - aug(i,:) * aug(j,i);
            end
        end
    end

    % Extract the solution
    x = aug(:, end);
end
```

Command Window

```
>> HW1_1_2_6
Solution for system (a):
     1
    -1

Solution for system (b):
     1
     4
    -1

Solution for system (c):
    0.5455
    0.4545

Solution for system (d):
    0.9600
    0.2800
    1.2000
```

System (d)

Equations:

$$x_1 - 2x_2 + 3x_3 + x_4 = 4$$
$$2x_1 + x_2 - x_3 + x_4 = 1$$
$$x_1 + 3x_2 + x_3 + x_4 = 3$$

Augmented Matrix:

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

Steps:

- Multiply Row 2 by 1/2 and subtract Row 1 from the new Row 2.

$$\left[\begin{array}{cccc|c} 1 & -2 & 3 & 1 & 4 \\ 0 & 2 & -\frac{5}{2} & \frac{1}{2} & -\frac{7}{2} \\ 1 & 3 & 1 & 1 & 3 \end{array}\right]$$

- Subtract Row 1 from Row 3.

$$\left[\begin{array}{cccc|c} 1 & -2 & 3 & 1 & 4 \\ 0 & 2 & -\frac{5}{2} & \frac{1}{2} & -\frac{7}{2} \\ 0 & 5 & -2 & 0 & -1 \end{array}\right]$$

- Multiply Row 3 by 2/5 and subtract 5/2 times the new Row 3 from Row 2.

$$\left[\begin{array}{cccc|c} 1 & -2 & 3 & 1 & 4 \\ 0 & 2 & -\frac{5}{2} & \frac{1}{2} & -\frac{7}{2} \\ 0 & 1 & -\frac{4}{5} & 0 & -\frac{2}{5} \end{array}\right]$$

- Multiply Row 3 by 2 and subtract 5 times Row 3 from Row 2 and add 2 times Row 3 to Row 1.

$$\left[\begin{array}{cccc|c} 1 & 0 & \frac{11}{5} & 1 & \frac{14}{5} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & -\frac{3}{2} \\ 0 & 1 & -\frac{4}{5} & 0 & -\frac{2}{5} \end{array}\right]$$

- Multiply Row 2 by 2 to get a leading one.

$$\left[\begin{array}{cccc|c} 1 & 0 & \frac{11}{5} & 1 & \frac{14}{5} \\ 0 & 0 & 1 & 1 & -3 \\ 0 & 1 & -\frac{4}{5} & 0 & -\frac{2}{5} \end{array}\right]$$

- Add 4/5 times Row 2 to Row 3 and subtract 11/5 times Row 2 from Row 1.

$$\left[\begin{array}{cccc|c} 1 & 0 & 0 & -\frac{2}{5} & \frac{32}{5} \\ 0 & 0 & 1 & 1 & -3 \\ 0 & 1 & 0 & \frac{4}{5} & -\frac{14}{5} \end{array}\right]$$

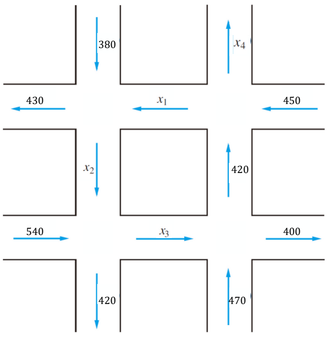
- Multiply Row 3 by 5/4 and add 2/5 times Row 3 to Row 1.

$$\left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 8 \\ 0 & 0 & 1 & 1 & -3 \\ 0 & 1 & 0 & 1 & -\frac{7}{2} \end{array}\right]$$

Solution:

$$x_1 = 8, x_2 = -\frac{7}{2}, x_3 = -3, x_4 = \frac{1}{2}$$

[1.2] ex. 15



To solve the system using Reduced Row Echelon Form (RREF), it first needs to set up the system of linear equations based on the traffic flows:

$$x_1 + 0x_2 + 0x_3 - x_4 = 450 - 380$$
$$x_1 - x_2 + 0x_3 + 0x_4 = -430$$
$$0x_1 + x_2 - x_3 + 0x_4 = 540 - 400$$
$$0x_1 + 0x_2 + x_3 + 0x_4 = 420 + 470$$

This translates to the following matrix equation:

$$AX = B$$

Now it augments matrix A with matrix B and perform row operations to bring it to RREF:

$$A = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \quad B = \begin{bmatrix} 70 \\ -430 \\ 140 \\ 890 \end{bmatrix}$$

Next, it performs the row operations step by step. The goal is to have only 1s on the diagonal and 0s everywhere else in the first 4 columns of the augmented matrix.

Step 1: Subtract Row 1 from Row 2.

Step 2: Add Row 2 to Row 3.

Step 3: No changes to Row 4, as it's already in the correct form for RREF.

After these operations, the matrix should look like this:

$$\left[\begin{array}{cccc|c} 1 & 0 & 0 & -1 & 70 \\ 1 & -1 & 0 & 0 & -430 \\ 0 & 1 & -1 & 0 & 140 \\ 0 & 0 & 1 & 0 & 890 \end{array}\right]$$

Step 4: Multiply Row 2 by -1 to get a leading 1.

Step 5: Add Row 3 to Row 2.

Step 6: Subtract Row 3 from Row 1.

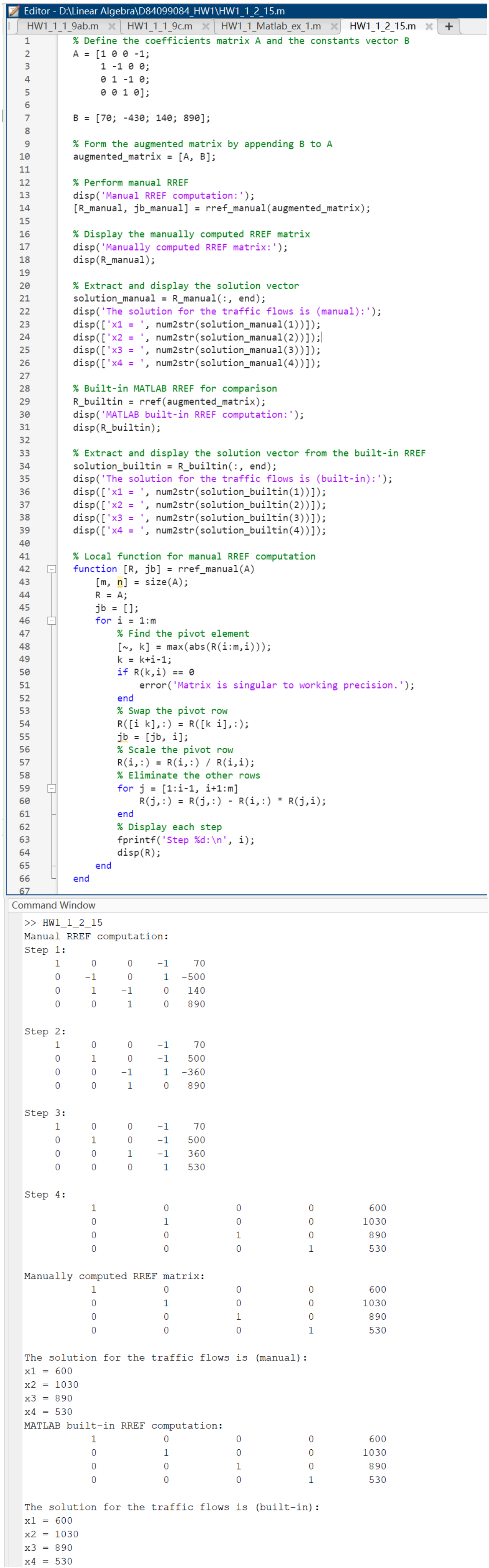
Step 7: Add Row 3 to Row 4.

The matrix should now look like this:

$$\left[\begin{array}{cccc|c} 1 & 0 & 0 & -1 & 70 \\ 0 & -1 & 0 & 1 & -500 \\ 0 & 0 & -1 & 1 & -360 \\ 0 & 0 & 1 & 0 & 890 \end{array}\right]$$

The final matrix is in RREF as the **solutions**:

$$x_1 = -290$$
$$x_2 = -860$$
$$x_3 = 890$$
$$x_4 = -360$$



Editor - D:\Linear Algebra\D84099084_HW1\HW1_1_2_15.m

HW1_1_1_9ab.m | HW1_1_1_9c.m | HW1_1_Matlab_ex_1.m | HW1_1_2_15.m

```matlab
% Define the coefficients matrix A and the constants vector B
A = [1 0 0 -1;
     1 -1 0 0;
     0 1 -1 0;
     0 0 1 0];

B = [70; -430; 140; 890];

% Form the augmented matrix by appending B to A
augmented_matrix = [A, B];

% Perform manual RREF
disp('Manual RREF computation:');
[R_manual, jb_manual] = rref_manual(augmented_matrix);

% Display the manually computed RREF matrix
disp('Manually computed RREF matrix:');
disp(R_manual);

% Extract and display the solution vector
solution_manual = R_manual(:, end);
disp('The solution for the traffic flows is (manual):');
disp(['x1 = ', num2str(solution_manual(1))]);
disp(['x2 = ', num2str(solution_manual(2))]);
disp(['x3 = ', num2str(solution_manual(3))]);
disp(['x4 = ', num2str(solution_manual(4))]);

% Built-in MATLAB RREF for comparison
R_builtin = rref(augmented_matrix);
disp('MATLAB built-in RREF computation:');
disp(R_builtin);

% Extract and display the solution vector from the built-in RREF
solution_builtin = R_builtin(:, end);
disp('The solution for the traffic flows is (built-in):');
disp(['x1 = ', num2str(solution_builtin(1))]);
disp(['x2 = ', num2str(solution_builtin(2))]);
disp(['x3 = ', num2str(solution_builtin(3))]);
disp(['x4 = ', num2str(solution_builtin(4))]);

% Local function for manual RREF computation
function [R, jb] = rref_manual(A)
    [m, n] = size(A);
    R = A;
    jb = [];
    for i = 1:m
        % Find the pivot element
        [~, k] = max(abs(R(i:m,i)));
        k = k+i-1;
        if R(k,i) == 0
            error('Matrix is singular to working precision.');
        end
        % Swap the pivot row
        R([i k],:) = R([k i],:);
        jb = [jb, i];
        % Scale the pivot row
        R(i,:) = R(i,:) / R(i,i);
        % Eliminate the other rows
        for j = [1:i-1, i+1:m]
            R(j,:) = R(j,:) - R(i,:) * R(j,i);
        end
        % Display each step
        fprintf('Step %d:\n', i);
        disp(R);
    end
end
```

Command Window

```
>> HW1_1_2_15
Manual RREF computation:
Step 1:
     1     0     0    -1    70
     0    -1     0     1  -500
     0     1    -1     0   140
     0     0     1     0   890

Step 2:
     1     0     0    -1    70
     0     1     0    -1   500
     0     0    -1     1  -360
     0     0     1     0   890

Step 3:
     1     0     0    -1    70
     0     1     0    -1   500
     0     0     1    -1   360
     0     0     0     1   530

Step 4:
           1           0           0           0         600
           0           1           0           0        1030
           0           0           1           0         890
           0           0           0           1         530

Manually computed RREF matrix:
           1           0           0           0         600
           0           1           0           0        1030
           0           0           1           0         890
           0           0           0           1         530

The solution for the traffic flows is (manual):
x1 = 600
x2 = 1030
x3 = 890
x4 = 530
MATLAB built-in RREF computation:
           1           0           0           0         600
           0           1           0           0        1030
           0           0           1           0         890
           0           0           0           1         530

The solution for the traffic flows is (built-in):
x1 = 600
x2 = 1030
x3 = 890
x4 = 530
```

[1.3] ex. 13

Given RREF matrix:

$$\begin{bmatrix} 1 & 2 & 0 & 3 & 1 & | & -2 \\ 0 & 0 & 1 & 2 & 4 & | & 5 \\ 0 & 0 & 0 & 0 & 0 & | & 0 \\ 0 & 0 & 0 & 0 & 0 & | & 0 \end{bmatrix}$$

and Vectors:

$$a_1 = \begin{bmatrix} 1 \\ 1 \\ 3 \\ 4 \end{bmatrix}, \quad a_3 = \begin{bmatrix} 2 \\ -1 \\ 1 \\ 3 \end{bmatrix}$$

(a) Find all solutions to the system.

The given RREF matrix corresponds to the following system of equations:

$$x_1 + 2x_2 + 3x_4 + x_5 = -2$$
$$x_3 + 2x_4 + 4x_5 = 5$$

Variables x_2, x_4, and x_5 are free variables because they are not leading variables in any row. The general solution to the system can be expressed in terms of these free variables. Let's assign

$$x_2 = s, x_4 = t, \text{and } x_5 = u$$

where $s, t,$ and u can take any real value.

Then we can express x_1 and x_3 as follows:

$$x_1 = -2 - 2s - 3t - u$$
$$x_3 = 5 - 2t - 4u$$

The general solution to the system is then:

$$\mathbf{x} = \begin{bmatrix} -2 - 2s - 3t - u \\ s \\ 5 - 2t - 4u \\ t \\ u \end{bmatrix}$$

where $s, t,$ and u are real numbers.

(b)Determine b, if:

$$a_1 = \begin{bmatrix} 1 \\ 1 \\ 3 \\ 4 \end{bmatrix}, \quad a_3 = \begin{bmatrix} 2 \\ -1 \\ 1 \\ 3 \end{bmatrix}$$

To determine the vector b, we use the vectors a_1 and a_3 which correspond to columns 1 and 3 of the original coefficient the matrix A. Since we have the RREF of A and the specific columns a_1 and a_3, we can back-substitute to find the corresponding entries in b.

For a_1 being the first column, represents x_1, and since the RREF shows that x_1 is a leading variable corresponding to the first equation, its contribution to b is just its coefficient from the equation:

$$x_1 + 2x_2 + 3x_4 + x_5 = -2$$

For a_3, it represents x_3, and as x_3 is a leading variable corresponding to the second equation, its contribution to b is the constant term of that equation, which is 5.

Hence, b would have its first entry as −2 and its third entry as 5, with the rest being zero, based on the provided RREF matrix. Thus, the vector b would be:

$$b = \begin{bmatrix} -2 \\ 0 \\ 5 \\ 0 \\ 0 \end{bmatrix}$$

Editor - D:\Linear Algebra\D84099084_HW1\HW1_1_3_13.m

HW1_1_1_9ab.m | HW1_1_1_9c.m | HW1_1_Matlab_ex_1.m | HW1_1_2_15.m | HW1_1_2_6.m | HW1_1_3_13.m

```matlab
% Define the RREF matrix
RREF = [1 2 0 3 1 -2;
        0 0 1 2 4 5;
        0 0 0 0 0 0;
        0 0 0 0 0 0];

% Identify the number of free variables
% These are the columns without leading 1s
num_free_vars = sum(all(RREF(:, 1:end-1) == 0, 1));

% Initialize free variables s, t, and u
syms s t u real;

% Express x1, x2, x3, x4, x5 in terms of free variables s, t, u
x2 = s;
x4 = t;
x5 = u;
x1 = -2 - 2*x2 - 3*x4 - x5;
x3 = 5 - 2*x4 - 4*x5;

% General solution vector
general_solution = [x1; x2; x3; x4; x5];

% Display the general solution
disp('General solution of the system:');
disp(general_solution);

% Determine b based on a1 and a3
% Since a1 corresponds to x1 and a3 corresponds to x3
% and using the values from the RREF matrix
b = zeros(5, 1);
b(1) = RREF(1, end); % The entry from the first row, last column of RREF
b(3) = RREF(2, end); % The entry from the second row, last column of RREF

% Display b
disp('Vector b based on columns a1 and a3:');
disp(b);

```

Command Window

```
>> HW1_1_3_13
General solution of the system:
- 2*s - 3*t - u - 2
                  s
      5 - 4*u - 2*t
                  t
                  u

Vector b based on columns a1 and a3:
    -2
     0
     5
     0
     0
```

賴澌雨 D84099084_HW1 Shih-Yu, Lai

[1.1] ex. 9(c) Please use 2D plots to illustrate your answers in (a) and (b).

The function is designed to visually represent a system of two linear equations in two variables and to identify their relationship—whether they have a unique solution (the lines intersect at a point), no solution (the lines are parallel and do not intersect), or infinitely many solutions (the lines coincide).

Function Overview

- **Inputs**: The function accepts four parameters, m_1, m_2, b_1, and b_2, which represent the coefficients and constants of a system of equations of the form $-m_1x_1 + x_2 = b_1$ and $-m_2x_1 + x_2 = b_2$.

- **Operation**: It first defines a range of x_1 values to plot the lines represented by the equations. It then calculates corresponding x_2 values for each line and plots these two lines on a graph with x_1 on the horizontal axis and x_2 on the vertical axis.

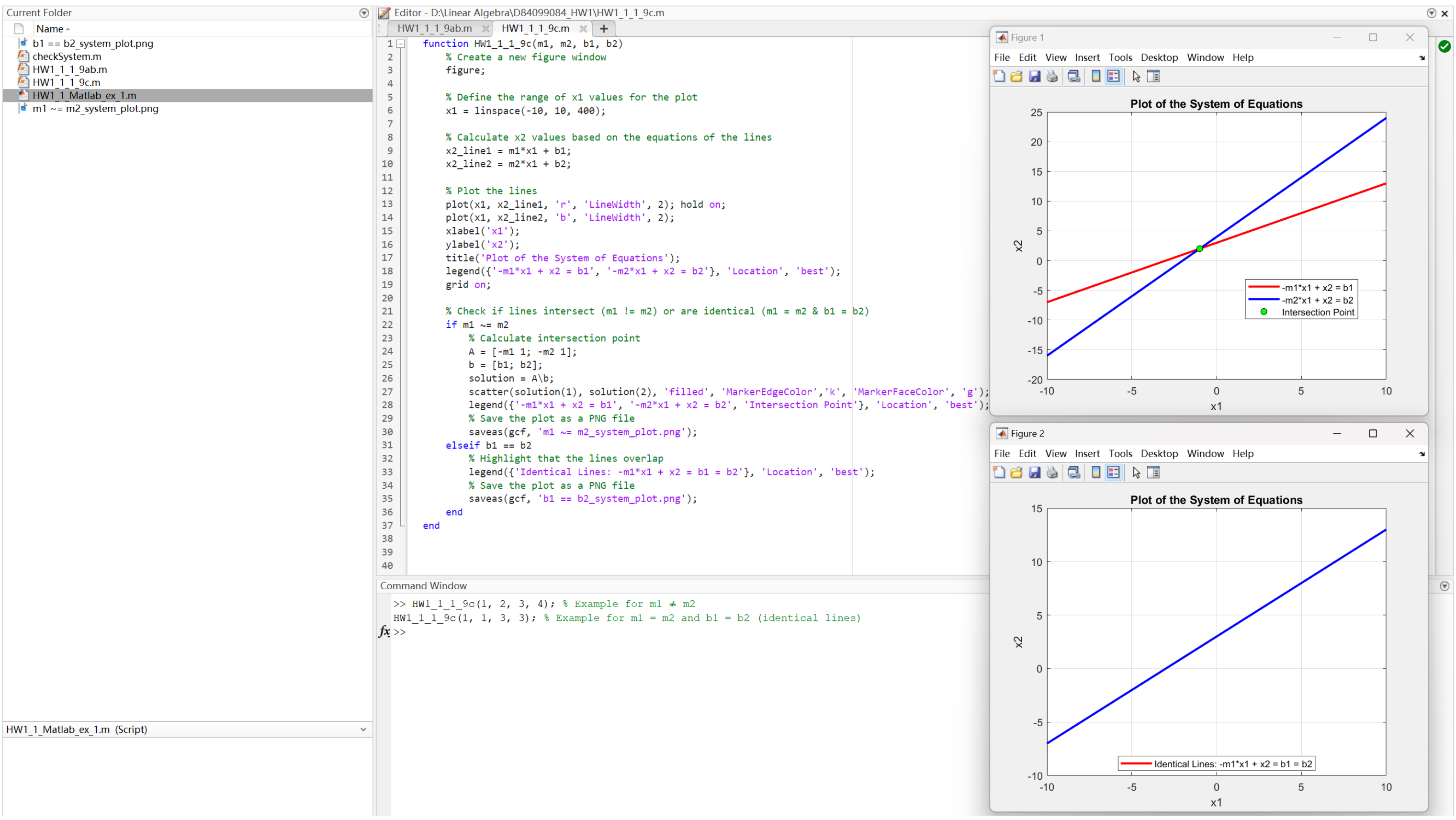
- **Unique Solution**: If $m_1 \neq m_2$, the lines will generally intersect at a single point, indicating a unique solution to the system. The function calculates this intersection point, highlights it on the plot, and updates the legend to indicate the presence of a unique intersection point.

- **No Solution or Infinitely Many Solutions**: If $m_1 = m_2$, the lines are either parallel (no solution) if $b_1 \neq b_2$, or coincide (infinitely many solutions) if $b_1 = b_2$. The script adjusts the legend accordingly to reflect whether the lines are identical or if a specific analysis was not included for parallel lines explicitly.

- **Visualization and Output**: The lines are displayed in different colors for clarity, and the plot includes a title, axis labels, and a legend to describe the elements of the plot. Finally, the plots are saved as a PNG file named in the current working directory.

How to Run

Execute the function with specific values for inputs to generate the plot and save it. For example, HW1_1_1_9c(1, 2, 3, 4) would plot the equations represented by those coefficients and constants, analyze their relationship, and save the plot.



賴澔雨 D84099084_HW1 Shih-Yu, Lai

[Matlab ex] ex. 1(a), (b). Please read introduction in the first paragraph of p.96. You may use round(10 * rand(5)) to randomly generate a 5 x 5 matrix.

For part (a):

- Generate matrices A and B.
- Compute
  $A1 = A * B, A2 = B * A, A3 = (A' * B')', A4 = (B' * A')'$.
- Check if
  $A1 = A2, A1 = A3, A1 = A4$

by computing their differences.

For part (b):

- Reuse matrices A and B.
- Compute
  $A1 = A. * B, A2 = A'. * B', A3 = (B. * A')', A4 = (B'. * A)$.
- Check if
  $A1 = A2, A1 = A3, A1 = A4$

by computing their differences.

Editor - D:\Linear Algebra\D84099084_HW1\HW1_1_Matlab_ex_1.m

HW1_1_1_9ab.m | HW1_1_1_9c.m | HW1_1_Matlab_ex_1.m

```matlab
% Generate random 5x5 matrices A and B
A = round(10 * rand(5));
B = round(10 * rand(5));

% Part (a)
A1 = A * B;
A2 = B * A;
A3 = (A' * B')';
A4 = (B' * A')';

% Check for equality by computing the difference
difference_a1_a2 = A1 - A2;
difference_a1_a3 = A1 - A3;
difference_a1_a4 = A1 - A4;

% Display the differences for part (a)
disp('Differences for part (a):');
disp('A1 - A2: ');
disp(difference_a1_a2);
disp('A1 - A3: ');
disp(difference_a1_a3);
disp('A1 - A4: ');
disp(difference_a1_a4);

% Part (b)
A1 = A .* B;
A2 = A' .* B';
A3 = (B .* A')';
A4 = (B' .* A);

% Check for equality by computing the difference
difference_b1_b2 = A1 - A2;
difference_b1_b3 = A1 - A3;
difference_b1_b4 = A1 - A4;

% Display the differences for part (b)
disp('Differences for part (b):');
disp('A1 - A2: ');
disp(difference_b1_b2);
disp('A1 - A3: ');
disp(difference_b1_b3);
disp('A1 - A4: ');
disp(difference_b1_b4);
```

Command Window

```
>> HW1_1_Matlab_ex_1
Differences for part (a):
A1 - A2:
    82   150   134   143    26
   -31    46    22    12   -69
   -34    20   -25    27   -87
   -42   -19   -13   -28   -32
   -44   -23   -30    34   -75

A1 - A3:
    82   150   134   143    26
   -31    46    22    12   -69
   -34    20   -25    27   -87
   -42   -19   -13   -28   -32
   -44   -23   -30    34   -75

A1 - A4:
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0

Differences for part (b):
A1 - A2:
     0     6     3    -2     3
    -6     0     6   -25     0
    -3    -6     0   -23   -14
     2    25    23     0   -27
    -3     0    14    27     0

A1 - A3:
     0    10     0   -50   -32
   -14     0     6   -20    -9
     0    -6     0     1    -8
    10    24    -7     0   -18
     4     6     9    27     0

A1 - A4:
     0    10     0   -50   -32
   -14     0     6   -20    -9
     0    -6     0     1    -8
    10    24    -7     0   -18
     4     6     9    27     0
```

Function Overview

Part (a)

- **Matrix Creation:**
  - A and B are generated by the expression round(10 * rand(5)).

- **Matrix Operations:**
  - $A1$ is determined by the matrix multiplication of A and B.
  - $A2$ results from the matrix multiplication of B and A.
  - $A3$ is the result of transposing A, multiplying by the transpose of B, and then transposing the product.
  - $A4$ is obtained by transposing B, multiplying by the transpose of A, and transposing the result.

Equality Check:

- The differences, and are computed to check for equality. If the matrices are equal, the difference will be a zero matrix.

Part (b)

- **Matrix Operations (Element-wise):**
  - $A1$ is calculated by the element-wise multiplication of A and B.
  - $A2$ is derived from the element-wise multiplication of the transposes of A and B.
  - $A3$ is the element-wise multiplication of B and the transposition of A, transposed.
  - $A4$ is computed by element-wise multiplying the transpose of B with A.
- **Equality Check:**
  - Similarly, $A1−A2$, $A1−A3$, and $A1−A4$ are calculated to verify if the resulting matrices are identical.