What Makes an Equation Linear?

All equations can be classified as either *linear* or *nonlinear*.

Linear Equations

A *linear* equation has only two types of terms: constants, and variables multiplied by constants.

constant

$$2x + 5 - y = 0$$

constant $*$ variable

Nonlinear Equations

Equations with any other type of term are *nonlinear* equations.

$$1 - z^2 = x$$

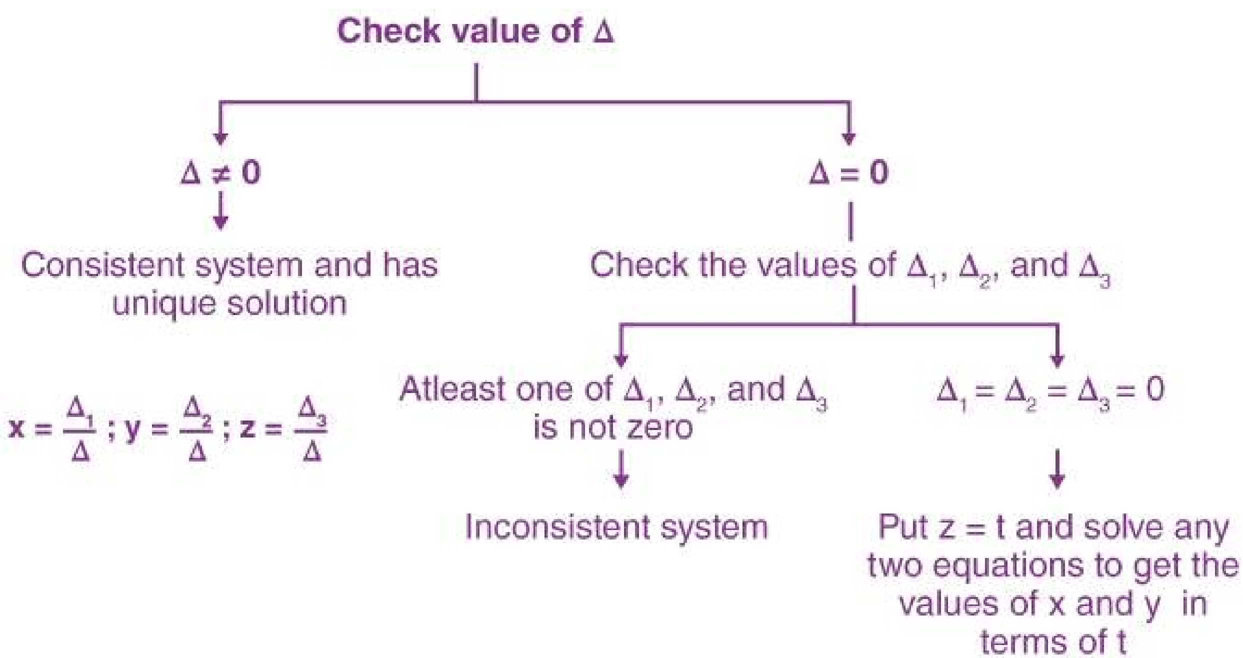
variable^2

$$e^x = 3$$

function of variable

$$\underbrace{y(1 - x)}_{\text{variable} \, * \, \text{variable}} = 2$$

MANUAL CHECKING OF THE TYPE OF SOLUTION



TYPES OF SOLUTION FOR LINEAR EQAUTION

1. **Unique Solution**: The planes intersect at a single point.
2. **Infinite Solutions**: The planes intersect along a line.
3. **No Solution**: The planes are parallel or offset with no common intersection.

- **Definition**: A system of linear equations can be represented as $Ax = b$.
  - **Matrix A**: Coefficients of the equations.
  - **Vector x**: Variables.
  - **Vector b**: Constants.
- **Example**:
  - Equations:

$$2x + 3y = 8$$

$$5x + y = 7$$

  - Matrix form:

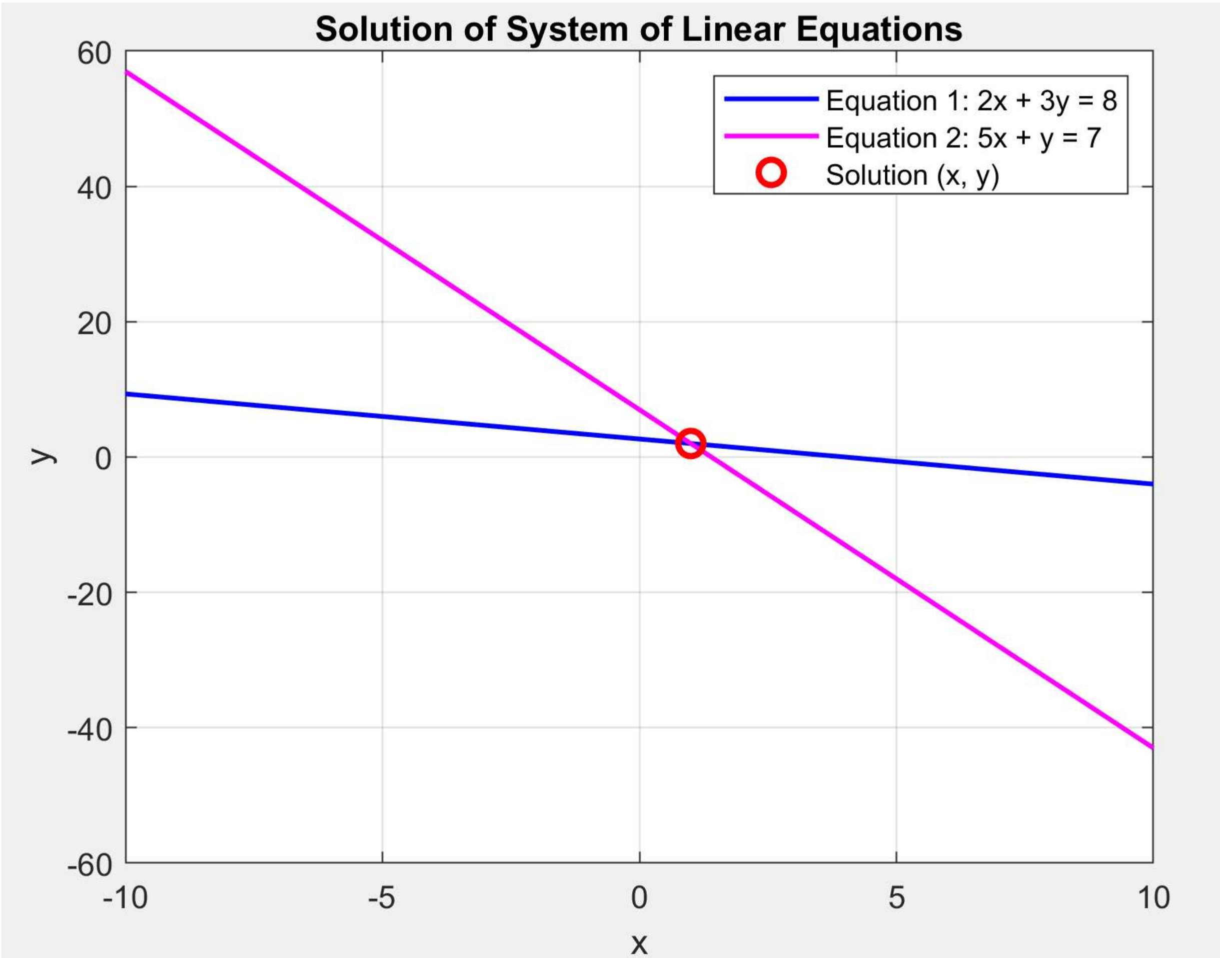
$$A = \begin{bmatrix} 2 & 3 \\ 5 & 1 \end{bmatrix}, \quad x = \begin{bmatrix} x \\ y \end{bmatrix}, \quad b = \begin{bmatrix} 8 \\ 7 \end{bmatrix}$$

```
% Define the larger matrix A and vector b
A = [1 2 3; 4 5 6; 7 8 10];
b = [9; 24; 45];

% Solve the system
x = A \ b;

% Display the solution
disp('Solution for x, y, and z:');
disp(x);
```

Left Division Operator (`\`) in MATLAB:

- The operator `\` is called the **left matrix division operator** in MATLAB.
- When you use `x = A \ b`, MATLAB is effectively solving the equation $Ax = b$ by calculating $x = A^{-1}b$ (if A is invertible).
- `A \ b` **means "solve the linear system $Ax = b$ for x"**.

```matlab
% Step 1: Define the matrix A and vector b
A = [2 3; 5 1];  % Matrix of coefficients
b = [8; 7];      % Vector of constants

% Step 2: Solve the system of linear equations Ax = b
x = A \ b;       % Solution for x and y

% Display the solution
disp('Solution for x and y:');
disp(x);

% Step 3: Define x values for plotting each equation as a line
x_vals = -10:0.1:10;  % Generate a range of x values from -10 to 10

% Step 4: Calculate corresponding y values for each equation
% Equation 1: 2x + 3y = 8 -> y = (8 - 2*x) / 3
y_vals_eq1 = (8 - 2 * x_vals) / 3;

% Equation 2: 5x + y = 7 -> y = (7 - 5*x)
y_vals_eq2 = (7 - 5 * x_vals);

% Step 5: Plot the equations as lines on a graph
figure;  % Create a new figure window
plot(x_vals, y_vals_eq1, 'b-', 'LineWidth', 1.5);  % Plot Equation 1 (blue line)
hold on;  % Hold the current plot to overlay Equation 2
plot(x_vals, y_vals_eq2, 'm-', 'LineWidth', 1.5);  % Plot Equation 2 (magenta line)

% Step 6: Plot the solution point
plot(x(1), x(2), 'ro', 'MarkerSize', 8, 'LineWidth', 2);  % Mark the solution point (x, y) in red

% Step 7: Add labels, title, and legend
xlabel('x');  % Label for x-axis
ylabel('y');  % Label for y-axis
title('Solution of System of Linear Equations');  % Title of the plot
legend('Equation 1: 2x + 3y = 8', 'Equation 2: 5x + y = 7', 'Solution (x, y)');  % Legend for plot

% Step 8: Display a grid for better visualization
grid on;  % Turn on the grid for the plot

% Step 9: Show the plot
hold off;  % Release the hold on the plot
```

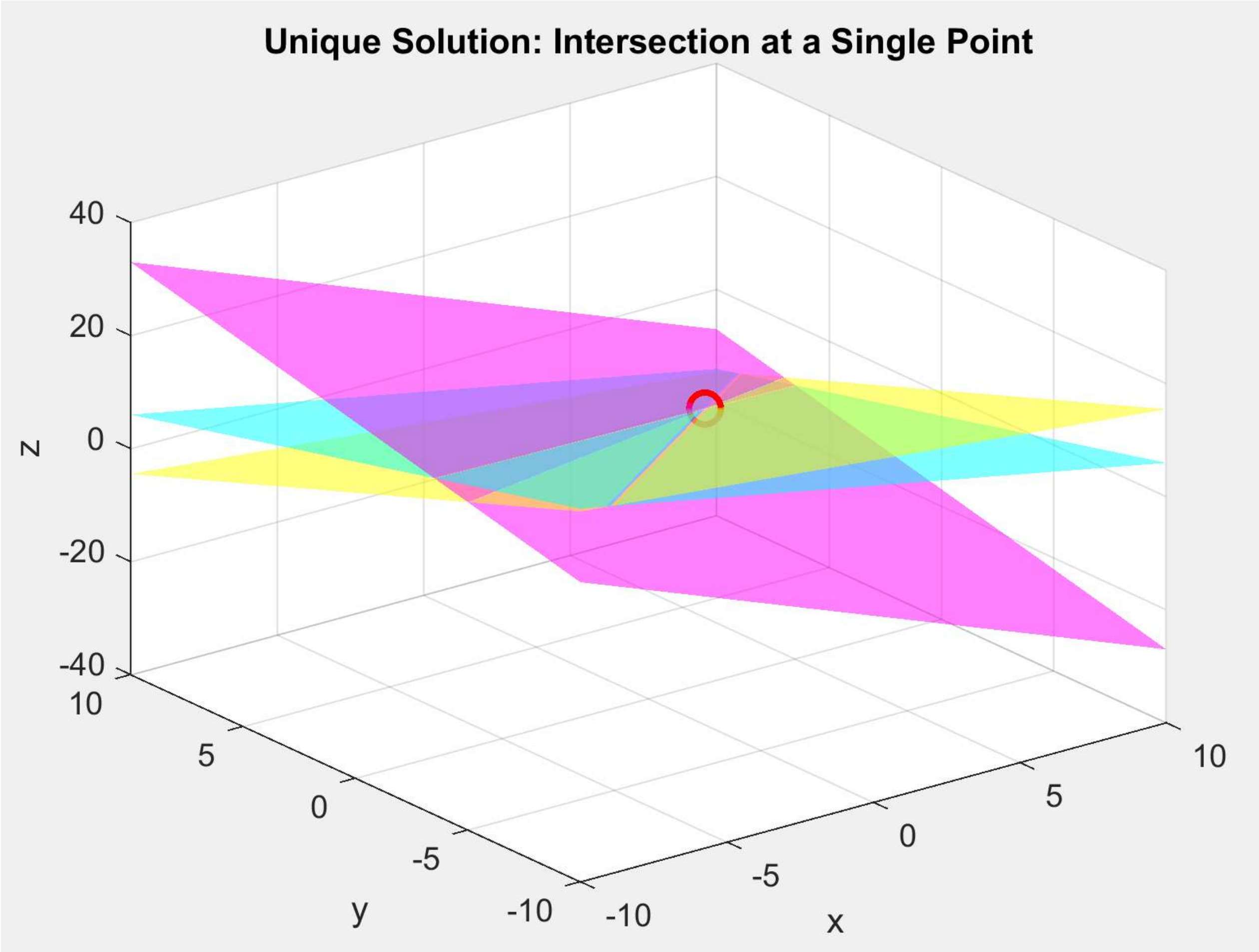

Solution of System of Linear Equations
Equation 1: 2x + 3y = 8
Equation 2: 5x + y = 7
Solution (x, y)
60
40
20
0
-20
-40
-60
y
-10
-5
0
5
10
x

In a 3x3 system, a unique solution occurs when the three planes intersect at a single point.

Example:

$$\begin{cases} x + y + z = 6 \\ 2x - y + z = 3 \\ x + 3y + 2z = 11 \end{cases}$$

```matlab
% Define the matrix A and vector b
A = [1 1 1; 2 -1 1; 1 3 2];
b = [6; 3; 11];

% Solve the system
x = A \ b;

% Plotting
[X, Y] = meshgrid(-10:1:10, -10:1:10);
Z1 = 6 - X - Y;
Z2 = (3 - 2*X + Y) / 1;
Z3 = (11 - X - 3*Y) / 2;

figure;
surf(X, Y, Z1, 'FaceAlpha', 0.5, 'EdgeColor', 'none', 'FaceColor', 'cyan');
hold on;
surf(X, Y, Z2, 'FaceAlpha', 0.5, 'EdgeColor', 'none', 'FaceColor', 'magenta');
surf(X, Y, Z3, 'FaceAlpha', 0.5, 'EdgeColor', 'none', 'FaceColor', 'yellow');
plot3(x(1), x(2), x(3), 'ro', 'MarkerSize', 10, 'LineWidth', 2);
xlabel('x'); ylabel('y'); zlabel('z');
title('Unique Solution: Intersection at a Single Point');
grid on;
```

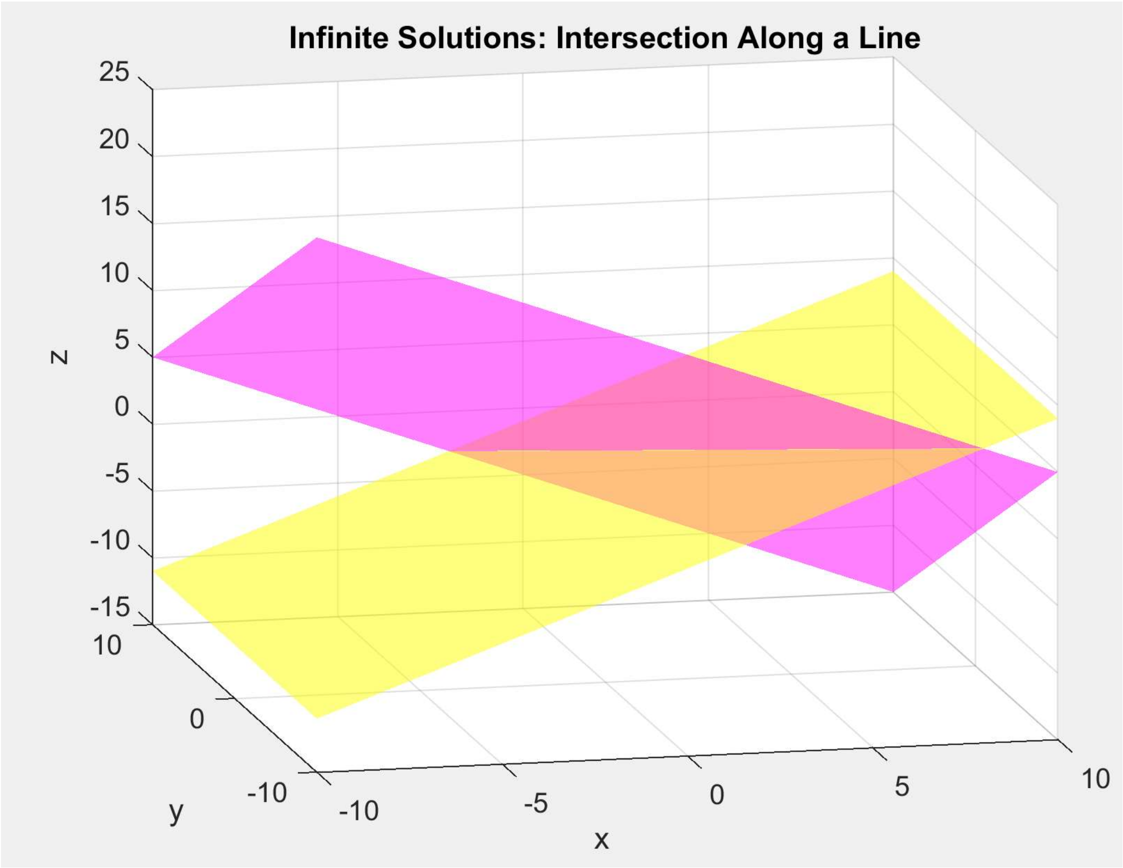

Unique Solution: Intersection at a Single Point
40
20
0
-20
-40
z
10
5
0
-5
-10
y
-10
-5
0
5
10
x

2. Infinite Solutions

In a 3x3 system, infinite solutions occur when the three planes intersect along a line. This situation happens when two or more planes are essentially the same (parallel and overlapping).

Example:

$$\begin{cases} x + y + z = 5 \\ 2x + 2y + 2z = 10 \\ x - y = 1 \end{cases}$$


Infinite Solutions: Intersection Along a Line
z
25
20
15
10
5
0
-5
-10
-15
y
10
0
-10
x
-10
-5
0
5
10

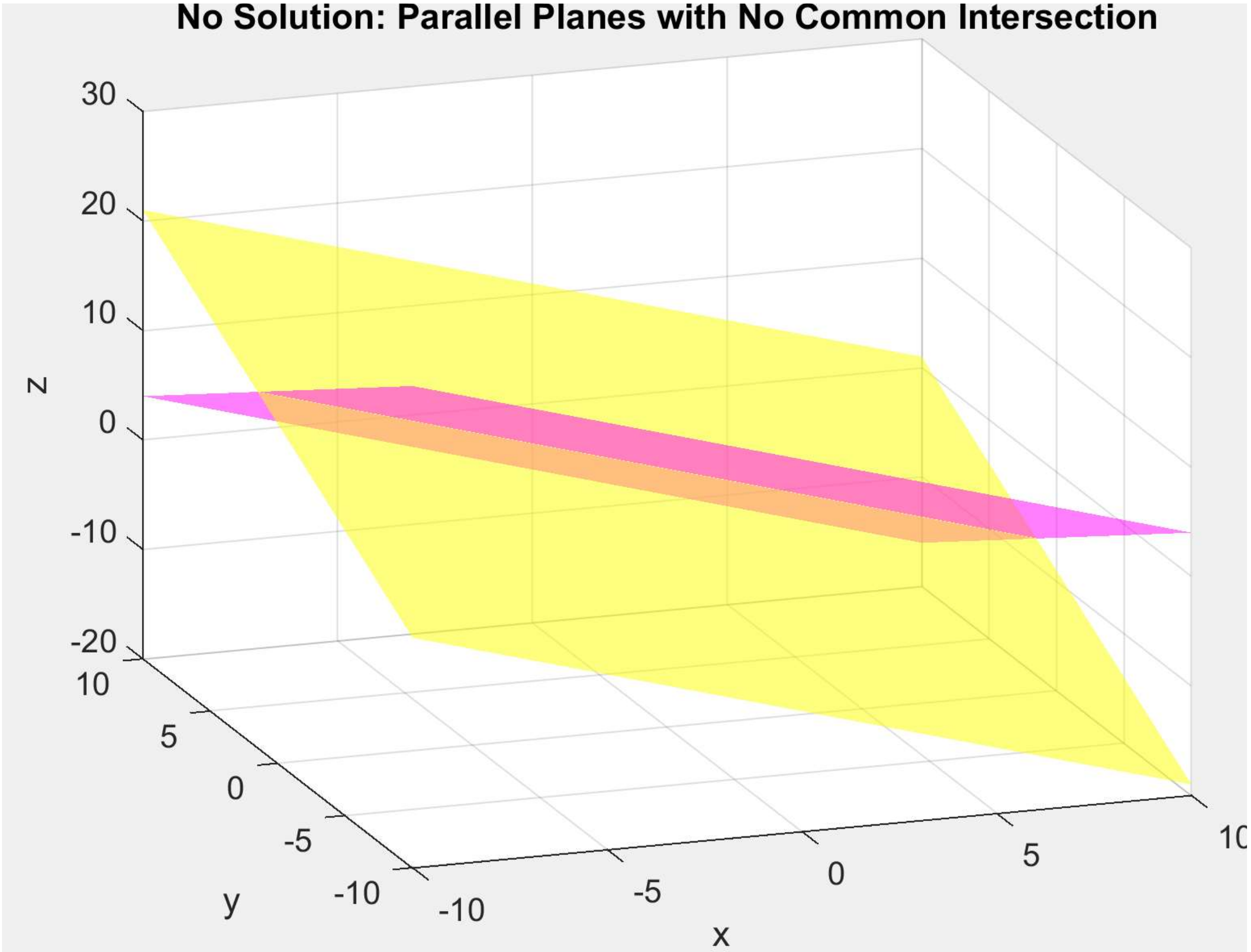
3. No Solution

In a 3x3 system, no solution occurs when the planes do not all intersect at a common point or line, indicating an inconsistency among the equations.

Example:

$$\begin{cases} x + y + z = 4 \\ 2x + 2y + 2z = 8 \\ x - y + z = 1 \end{cases}$$

```
% Define the matrix A and vector b
A = [1 1 1; 2 2 2; 1 -1 1];
b = [4; 8; 1];

% This system does not have a solution

% Plotting
[X, Y] = meshgrid(-10:1:10, -10:1:10);
Z1 = 4 - X - Y;
Z2 = (8 - 2*X - 2*Y) / 2;  % This is parallel to Z1 but offset
Z3 = (1 - X + Y);

figure;
surf(X, Y, Z1, 'FaceAlpha', 0.5, 'EdgeColor', 'none', 'FaceColor', 'cyan');
hold on;
surf(X, Y, Z2, 'FaceAlpha', 0.5, 'EdgeColor', 'none', 'FaceColor', 'magenta');  % Parallel but offset
surf(X, Y, Z3, 'FaceAlpha', 0.5, 'EdgeColor', 'none', 'FaceColor', 'yellow');
xlabel('x'); ylabel('y'); zlabel('z');
title('No Solution: Parallel Planes with No Common Intersection');
grid on;
```


No Solution: Parallel Planes with No Common Intersection
30
20
10
0
-10
-20
z
10
5
0
-5
-10
y
-10
-5
0
5
10
x