Gaussian Elimination Algorithm for Solving a System of Linear Equations in C++

Types of Solutions

1. Unique Solution: Exists if each leading 1 corresponds to a distinct variable (no free variables).
2. No Solution: Occurs if any row reduces to [0 0 0 | b] with $b \neq 0$, indicating inconsistency.
3. Infinitely Many Solutions: Exists if the row echelon form has a row of all zeros [0 0 0 | 0], implying free variables.

Algorithm

Input: An n × (n + 1) augmented matrix representing the system of equations. Output: Either the unique solution, a message indicating no solution, or an expression in terms of free variables for infinitely many solutions.

---

1. Forward Elimination:
    - For each row i from 1 to n − 1:
        - Leading Selection:
            - Swap the current row i with the row below it that has the largest absolute value in the i-th column (for numerical stability).
            - Print Operation: "Swapped row i with row containing the largest value in column i".
            - Print Matrix after the swap.
        - Normalize Leading:
            - Divide each element in row i by A[i][i] to make the leading 1.

        - Print Operation: "Normalized row i by dividing by A[i][i]".
        - Print Matrix after normalization.
    - Eliminate Below:
        - For each row j > i:
            - Calculate $f = \frac{A[j][i]}{A[i][i]}$
            - Subtract f × (row i) from row j to make A[j][i] = 0.
            - Print Operation: "Row j – f × row i".
            - Print Matrix after each elimination.
2. Check for Special Cases:
    - No Solution: If any row has the form [0 0 0 | b] with b ≠ 0, print "No solution exists" and exit.
    - Infinite Solutions: If the row echelon form has a row of all zeros [0 0 0 | 0], indicate infinitely many solutions and set free variables.
3. Backward Substitution:
    - For each row i from n down to 1:
        - Solve for the variable in row i by isolating it and using values from subsequent rows.
        - Substitute back into previous rows to simplify.
        - Print Operation: "Solved for variable in row i".
        - Print Matrix after each substitution.
4. Output:
    - Unique Solution: Output each variable's value.
    - Infinite Solutions: Output the solution in terms of free variables.
    - No Solution: Already handled in Step 2 if encountered.