伪代码 test

test

2024 年 10 月 11 日

Algorithm 1: Main Routine for Gaussian Elimination Solver

```
Input: Input File Path (string), tol (long double), max_iter (int)
Output: Solutions (array)
while True do
    selected_file ← SelectInputFile() ;                                  // Select the input file
    if selected_file is empty then
        exit ;                                                           // Exit if no file is selected
    end
    start_time ← StartTimer() ;                                          // Start the timer
    InitMatrix(matrix, selected_file, rows, cols) ;                      // Initialize the matrix
    ShowEquations(matrix, rows, cols) ;                                  // Display the system of equations
    exchange_count ← GaussianElimination(matrix, rows, cols) ;           // Perform Gaussian elimination
    rank ← DetermineRank(matrix, rows, cols) ;                           // Determine the rank of the matrix
    consistent ← CheckConsistency(matrix, rows, cols) ;                  // Check if the system is consistent
    if not consistent then
        DisplaySolution("No solution") ;                                 // Display no solution message
    end
    else if rank < (cols − 1) then
        ShowGeneralSolution(matrix, rows, cols, rank) ;                  // Display parameterized solution
    end
    else
        solution ← BackSubstitution(matrix, rows, cols, solution) ;      // Perform back substitution
        if solvable then
            DisplaySolution(solution) ;                                  // Display the unique solution
        end
        else
            DisplaySolution("No solution") ;                             // Display no solution if back substitution fails
        end
    end
    StopTimer(start_time) ;                                              // Stop the timer
    choice ← AskRunAgain() ;                                             // Ask if the user wants to run again
    if choice ≠ 'y' and choice ≠ 'Y' then
        break ;                                                          // Exit loop if the choice is not 'y' or 'Y'
    end
end
WaitForExit() ;                                                          // Wait for program exit
```

Algorithm 2: Pivoting to Select the Maximum Element in a Column

```
Input: matrix (Matrix), current_row (int), total_rows (int)
Output: imax (int)
1  imax ← current_row;
2  max_val ← |matrix[current_row][current_row]|;
3  for $i$ ← $current\_row + 1$ to $total\_rows - 1$ do
4      val ← |matrix[$i$][current_row]|;
5      if $val > max\_val$ then
6          imax ← $i$;
7          max_val ← val;
8      end
9  end
10 return imax;
```