

# Programming Assignment 6: Linear Equation System Solver

---

D1262028 李皓鈞

## Introduction

This report discusses the development and implementation of a C++ project designed to solve a system of linear equations. The project includes classes for matrices and vectors with class inheritance, and implements an application program to solve linear equation systems. The main components of the project are a base class Matrix and two derived classes Vector and SMatrix, which represent vectors and square matrices.

## Project Structure

The project directory is named Assignment6\_D1262028 and includes the following files:

- matrix.h: Header file for the Matrix class
- matrix.cpp: Implementation file for the Matrix class
- smatrix.h: Header file for the SMatrix class
- smatrix.cpp: Implementation file for the SMatrix class
- vector.h: Header file for the Vector class
- vector.cpp: Implementation file for the Vector class
- linear\_equation\_system\_solver.cpp: Application program to solve linear equation systems

## Implementation

### Header Files

The header files define the classes and their member functions. The Matrix class serves as the base class, while the Vector and SMatrix classes are derived from it. These classes include necessary operations for handling matrices and vectors.

### Source Files

The source files implement the member functions of each class. The matrix.cpp file implements the Matrix class methods, smatrix.cpp implements the SMatrix class methods, and vector.cpp implements the Vector class methods. These implementations include functionality such as matrix multiplication, solving linear systems, and printing matrices and vectors.

## Application Program

The application program, `linear_equation_system_solver.cpp`, contains the main logic for solving the system of linear equations. It performs the following steps:

1. Inputs the rank of the linear equation system.
2. Randomly generates elements for the coefficient matrix A and the constant vector C.
3. Prints the coefficient matrix A, constant vector C, and the system of linear equations.
4. Solves the linear equation system to obtain the solution vector X.
5. Prints the solution vector X.
6. Verifies the solution by computing  $AX - C$  and checking if it is the zero vector within a specified error tolerance.

## Example

Below is an example implementation of the main logic in `linear_equation_system_solver.cpp`:

```
#include <iostream>
#include <vector>
#include <cstdlib>
#include <ctime>
#include <iomanip>
#include "matrix.h"
#include "smatrix.h"
#include "vector.h"

using namespace std;

int main() {
    srand(time(0)); // Seed for random number generation

    int n;
    cout << "Enter the rank of the linear equation system (0 < n ≤ 10): ";
    cin >> n;

    if (n <= 0 || n > 10) {
        cerr << "Invalid rank, must be between 1 and 10." << endl;
        return -1;
    }

    SMatrix A(n);
    Vector C(n);

    // Randomly generate elements for A and C
```

```

for (int i = 0; i < n; ++i) {
    C[i] = static_cast<float>(rand()) / RAND_MAX;
    for (int j = 0; j < n; ++j) {
        A[i, j] = static_cast<float>(rand()) / RAND_MAX;
    }
}

// Print the coefficient matrix A and constant vector C
cout << "Coefficient Matrix A:" << endl;
A.print();
cout << "Constant Vector C:" << endl;
C.print();

// Solve the system and obtain solution vector X
Vector X = A.solve(C);

// Print the solution vector X
cout << "Solution Vector X:" << endl;
X.print();

// Verify the solution
Vector AX = A * X;
Vector error = AX - C;

cout << "Verification (AX - C):" << endl;
error.print();

bool correct = true;
for (int i = 0; i < n; ++i) {
    if (fabs(error[i]) > 1e-6) {
        correct = false;
        break;
    }
}

if (correct) {
    cout << "Solution verified successfully!" << endl;
} else {
    cout << "Solution verification failed!" << endl;
}

return 0;

```

}

## Conclusion

The development of this project involved creating classes for matrices and vectors, implementing their methods, and solving a system of linear equations. The program successfully reads input, generates random matrices and vectors, solves the system, and verifies the solution. This assignment provided valuable experience in object-oriented programming and numerical methods for solving linear equations.