Отчёт по коду для решения СЛАУ методом Гаусса

Введение

Метод Гаусса, или метод Гаусса-Жордана, представляет собой алгоритм решения систем линейных алгебраических уравнений. Этот метод заключается в приведении системы к верхнетреугольному виду с последующим решением полученной системы уравнений.

Код программы

```
#include <stdio.h>
#include <stdlib.h>

#define N 3 // Размерность системы

void printMatrix(double matrix[N][N+1]) {
   for (int i = 0; i < N; i++) {
      for (int j = 0; j <= N; j++) {
         printf("%10.3lf ", matrix[i][j]);
      }
      printf("\n");
   }
}

void gaussElimination(double matrix[N][N+1]) {
   int i, j, k;

   for (i = 0; i < N; i++) {
      // Поиск главного элемента
      double max = abs(matrix[i][i]);
      int maxRow = i;
      for (k = i + 1; k < N; k++) {
         if (abs(matrix[k][i]) > max) {
            max = abs(matrix[k][i]);
            maxRow = k;
         }
      }

      // Меняем текущую строку с главной строкой
      for (k = i; k <= N; k++) {
         double tmp = matrix[maxRow][k];
         matrix[maxRow][k] = matrix[i][k];
         matrix[i][k] = tmp;
      }

      // Приведение к верхнетреугольному виду
      for (j = i + 1; j < N; j++) {
         double factor = -matrix[j][i] / matrix[i][i];
         for (k = i; k <= N; k++) {
            if (i == k)
               matrix[j][k] = 0;
            else
               matrix[j][k] += factor * matrix[i][k];
         }
      }
   }

   // Обратный ход метода Гаусса
   double x[N];
   for (i = N - 1; i >= 0; i--) {
      x[i] = matrix[i][N] / matrix[i][i];
```

```
        for (j = i - 1; j >= 0; j--) {
            matrix[j][N] -= matrix[j][i] * x[i];
        }
    }

    // Вывод решения
    printf("Решение:\n");
    for (i = 0; i < N; i++) {
        printf("x%d = %10.3lf\n", i + 1, x[i]);
    }
}

int main() {
    // Пример матрицы для теста
    double matrix[N][N+1] = {
        {3, -0.1, -0.2, 7.85},
        {0.1, 7, -0.3, -19.3},
        {0.3, -0.2, 10, 71.4}
    };

    printf("Исходная матрица:\n");
    printMatrix(matrix);

    gaussElimination(matrix);

    return 0;
}
```

Объяснение кода

Заголовочные файлы:
#include <stdio.h>
#include <stdlib.h>

Определение размера системы:
#define N 3 // Размерность системы

Макрос N определяет размерность системы уравнений. В данном примере она равна 3.

Функция printMatrix**:**

```
void printMatrix(double matrix[N][N+1]) {

    for (int i = 0; i < N; i++) {

        for (int j = 0; j <= N; j++) {

            printf("%10.3lf ", matrix[i][j]);

        }

        printf("\n");

    }
}
```

Эта функция выводит матрицу на экран для наглядности. Она принимает

двумерный массив matrix и выводит его элементы в виде таблицы.

Функция gaussElimination**:**

```
void gaussElimination(double matrix[N][N+1]) {

  int i, j, k;



  for (i = 0; i < N; i++) {

    // Поиск главного элемента

    double max = abs(matrix[i][i]);

    int maxRow = i;

    for (k = i + 1; k < N; k++) {

      if (abs(matrix[k][i]) > max) {

        max = abs(matrix[k][i]);

        maxRow = k;

      }

    }



    // Меняем текущую строку с главной строкой

    for (k = i; k <= N; k++) {

      double tmp = matrix[maxRow][k];

      matrix[maxRow][k] = matrix[i][k];

      matrix[i][k] = tmp;

    }



    // Приведение к верхнетреугольному виду

    for (j = i + 1; j < N; j++) {

      double factor = -matrix[j][i] / matrix[i][i];

      for (k = i; k <= N; k++) {

        if (i == k)
```

```
                matrix[j][k] = 0;
            else
                matrix[j][k] += factor * matrix[i][k];
        }
    }
}

// Обратный ход метода Гаусса
double x[N];
for (i = N - 1; i >= 0; i--) {
    x[i] = matrix[i][N] / matrix[i][i];
    for (j = i - 1; j >= 0; j--) {
        matrix[j][N] -= matrix[j][i] * x[i];
    }
}

// Вывод решения
printf("Решение:\n");
for (i = 0; i < N; i++) {
    printf("x%d = %10.3lf\n", i + 1, x[i]);
}
}
```

Эта функция реализует метод Гаусса. Она включает два основных этапа:

Прямой ход:
Поиск главного элемента в каждом столбце для уменьшения ошибок округления.
Перестановка строк так, чтобы главный элемент оказался на диагонали.
Приведение системы к верхнетреугольному виду путём исключения переменных.

Обратный ход:

Вычисление значений переменных, начиная с последней строки и продвигаясь вверх.

Функция main**:**

```
int main() {

  // Пример матрицы для теста

  double matrix[N][N+1] = {

    {3, -0.1, -0.2, 7.85},

    {0.1, 7, -0.3, -19.3},

    {0.3, -0.2, 10, 71.4}

  };



  printf("Исходная матрица:\n");

  printMatrix(matrix);



  gaussElimination(matrix);



  return 0;

}
```

Эта функция:

Инициализирует матрицу коэффициентов и столбец свободных членов.
Выводит исходную матрицу на экран.
Вызывает функцию gaussElimination для решения системы уравнений.
Выводит найденные значения переменных.