

Университет ИТМО  
Факультет программной инженерии и компьютерной техники

**Вычислительная математика.**  
**Лабораторная работа №1.**  
**Решение систем линейных уравнений**

Группа: Р32131  
Студент: Смирнов Виктор Игоревич

2023

# Ключевые слова

Системы линейных алгебраических уравнений, СЛАУ, алгоритмы, вычислительные методы.

## 1 Цель работы

Целью проделанной работы было прежде всего ознакомление с вычислительными методами решения систем линейных алгебраических уравнений. Задача состояла в том, чтобы реализовать необходимые алгоритмы на языке программирования, а также разработать программное приложение с пользовательским интерфейсом.

## 2 Описание метода

Был реализован метод главных элементов ([1], с. 281).

## 3 Реализация алгоритма

```
1 namespace math::eq::lin::sys::solve::gauss {
2
3 struct index_pair { // NOLINT
4     size_t i;
5     size_t j;
6 };
7
8 template <typename F, size_t N>
9 static index_pair
10 peek(const linal::matrix<F, N, N>& matrix, index_pair start) {
11     auto max = start;
12     size_t j = start.j; // NOLINT
13     for (size_t i = start.i; i < N; i++) { // NOLINT
14         if (matrix[max.i][max.j] < matrix[i][j]) {
15             max.i = i;
16             max.j = j;
17         }
18     }
19     return max;
20 }
21
22 template <size_t N>
23 struct triangulization_result { // NOLINT
24     std::array<index_pair, N> swaps;
25     size_t col_swaps_count = 0;
26     size_t row_swaps_count = 0;
27 };
28
29 template <typename F, size_t N>
30 triangulization_result<N> triangulate(lineqsys<F, N>& sys) {
31     triangulization_result<N> result = {};
32
33     for (size_t row = 0; row < N; row++) {
34         // Put max element in working position
35         // const index_pair pos = {row, row};
36         const auto pos = peek<F, N>(sys.a, {row, row});
37
38         if (sys.a[pos.i][pos.j] == 0) {
39             throw std::invalid_argument("matrix can't be solved by gauss");
40         }
41
42         if (pos.i != row) {
43             sys.a.swap_rows(pos.i, row);
44             std::swap(sys.b[pos.i], sys.b[row]);
45             result.row_swaps_count++;
46         }
47
48         if (pos.j != row) {
49             sys.a.swap_cols(pos.j, row);
50             result.swaps[result.col_swaps_count++] = {.i = pos.j, .j = row};
```

```

51     }
52
53     // Go ahead and make zeros below
54     for (size_t i = row + 1; i < N; i++) { // NOLINT
55         F multiplier = sys.a[i][row] / sys.a[row][row];
56         sys.a[i] -= multiplier * sys.a[row];
57         sys.b[i] -= multiplier * sys.b[row];
58     }
59 }
60
61 return result;
62 }
63
64 template <typename F, size_t N>
65 static linal::vector<F, N> solve_triangle(const lineqsys<F, N>& sys) {
66     linal::vector<F, N> result; // NOLINT: fill in the loop
67     for (size_t row = N - 1; row--> 0) {
68         auto x = sys.b[row]; // NOLINT
69         for (size_t col = row + 1; col < N; col++) { // NOLINT
70             x -= sys.a[row][col] * result[col];
71         }
72         result[row] = x / sys.a[row][row];
73         if (row == 0) {
74             break;
75         }
76     }
77     return result;
78 }
79
80 template <typename F, size_t N>
81 struct result : solution<F, N> { // NOLINT
82 public:
83     result(
84         math::linal::vector<F, N> value, // NOLINT
85         math::linal::vector<F, N> error, // NOLINT
86         const lineqsys<F, N>& triangle,
87         F det
88     )
89         : solution<F, N>(value, error), triangle(triangle), det(det) {}
90
91     lineqsys<F, N> triangle; // NOLINT
92     F det; // NOLINT
93 };
94
95 template <typename F, size_t N>
96 result<F, N> solve(const lineqsys<F, N>& sys) {
97     auto triangle = sys;
98     auto tresult = triangulate(triangle);
99     auto value = solve_triangle(triangle);
100
101     if (tresult.col_swaps_count > 0) {
102         for (size_t i = tresult.col_swaps_count - 1; i--> 0) { // NOLINT
103             auto swap = tresult.swaps[i];
104             std::swap(value[swap.i], value[swap.j]);
105             if (i == 0) {
106                 break;
107             }
108         }
109     }
110
111     F det = 1;
112     for (size_t i = 0; i < N; i++) {
113         det *= triangle.a[i][i];
114     }
115     if ((tresult.row_swaps_count + tresult.col_swaps_count) % 2 == 1) {
116         det *= -1;
117     }
118
119     return result<F, N>(
120         value, //
121         sys.b - sys.a * value, //
122         triangle, //
123         det //

```

```

124 );
125 }
126
127 }

```

Листинг 1: Реализация метода главных элементов на языке C++

## 4 Примеры использования программного приложения

```

1 $ cat a.txt
2 3
3 6 -1 -1 11.33
4 -1 6 -1 32
5 -1 -1 6 42
6
7 $ math-tool lineqsys a.txt
8 lineqsys gauss method results report
9 det: 196
10 triangle matrix:
11      6      -1      -1 | 11.33
12      0    5.833    -1.167 | 33.89
13      0      0      5.6 | 50.67
14 result: { 4.666, 7.619, 9.047 }
15 error:  { -2.861e-06, 3.815e-06, 3.815e-06 }

```

Листинг 2: Пример использования приложения 1

```

1 $ cat null.txt
2 5
3 0 0 0 0 0 0
4 0 0 0 0 0 0
5 0 0 0 0 0 0
6 0 0 0 0 0 0
7 0 0 0 0 0 0
8
9 $ math-tool lineqsys null.txt
10 error: matrix can't be solved by gauss

```

Листинг 3: Пример использования приложения 2

## 5 Вывод

В процессе выполнения данной лабораторной работы я ознакомился с некоторыми вычислительными методами для решения систем линейных алгебраических уравнений. Было очень интересно и увлекательно реализовывать их на языке C++.

## Список литературы

[1] Б.П. Демидович, И.А. Марон Основы вычислительной математики: учебное пособие — 1966 год.