

Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №1

по «Вычислительной математике»

«Решение систем линейных алгебраических уравнений»

Выполнил:

Студент группы Р32312

Лебедев В.В.

Преподаватель:

Перл О.В.

Санкт-Петербург

2023

Описание метода

Метод простых итераций - численный метод решения систем линейных алгебраических уравнений. Метод позволяет получить значения корней системы с заданной точностью ϵ .

СЛАУ задается в виде: $AX = B$,

где A - матрица коэффициентов, X - вектор неизвестных, B - вектор свободных членов.

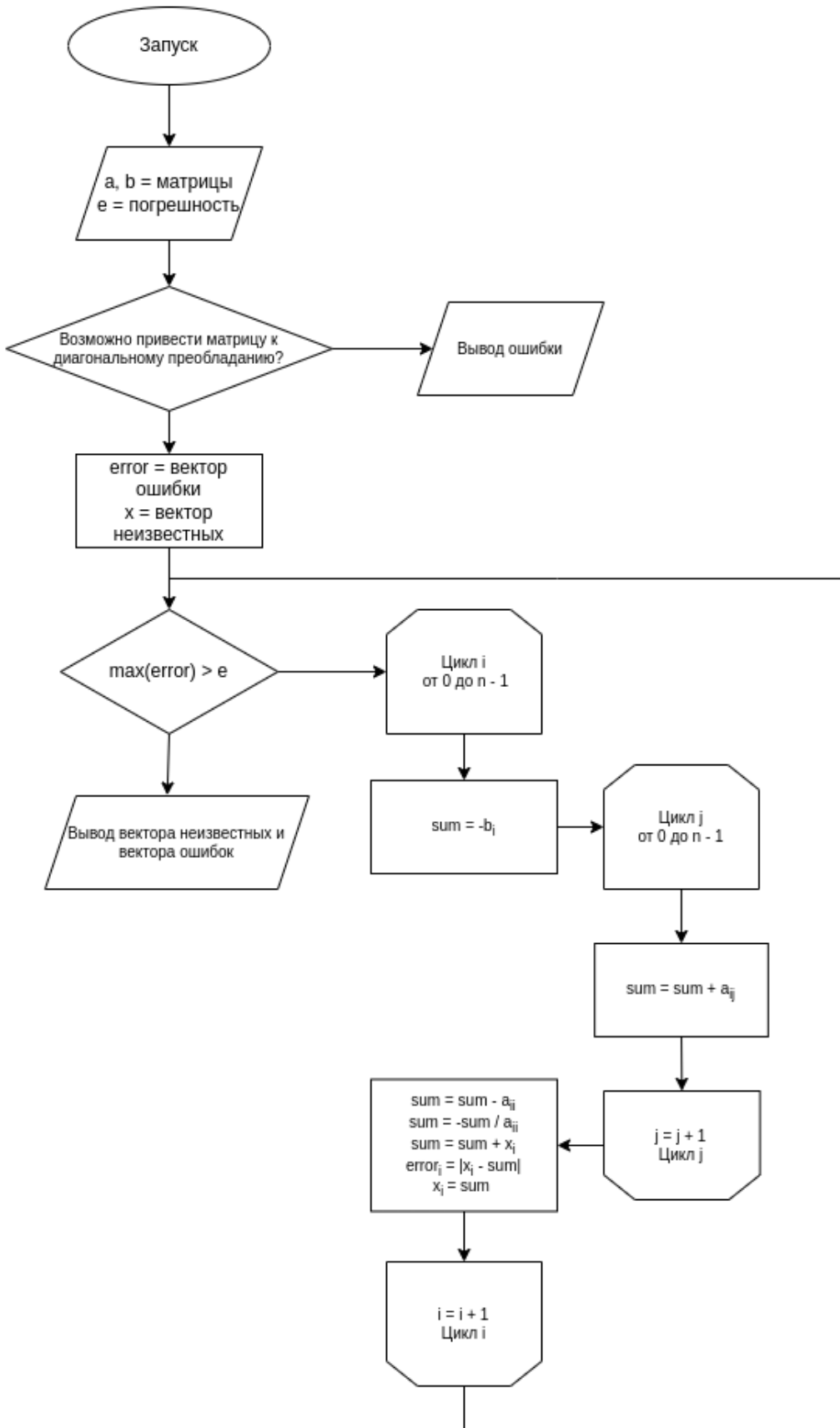
В итоге метод сводится к последовательным итерациям по j :

$$x_i^{j+1} = x_i^j - \frac{1}{a_{ii}} \left(\sum_{k=1}^n a_{ik} x_k^j - b_i \right),$$
 где для каждой строки i вычисляется новое значение x_i .

При этом, сходимость обеспечивается при условии диагонального преобладания

матрицы коэффициентов: $|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|$.

Блок-схема численного метода



Листинг программы

Приведение к диагональному преобладанию:

```
dominant_type is_dominant_row(std::vector<double> row, std::size_t col) {
    double sum = 0;
    for (std::size_t i = 0; i < row.size() - 1; i++) {
        sum += std::abs(x: row[i]);
    }
    sum -= std::abs(x: row[col]);
    if (std::abs(x: row[col]) > sum) {
        return STRICTLY_DOMINANT;
    } else if (std::abs(x: row[col]) >= sum) {
        return DOMINANT;
    } else {
        return NONDOMINANT;
    }
}
```

```
bool diagonal_dominance(std::vector<std::vector<double>>& matrix) {
    bool strict_flag = false;
    for (std::size_t col = 0; col < matrix.size(); col++) {
        bool dominant_flag = false;
        for (std::size_t row = col; row < matrix.size(); row++) {
            dominant_type dominant = is_dominant_row(row: matrix[row], col);
            if (dominant == STRICTLY_DOMINANT) {
                strict_flag = true;
            }
            if (dominant == STRICTLY_DOMINANT || dominant == DOMINANT) {
                std::swap(&: matrix[row], &: matrix[col]);
                dominant_flag = true;
                break;
            }
        }
        if (!dominant_flag) {
            return false;
        }
    }
    return strict_flag;
}
```

Итерационный метод:

```
static double iterate(std::vector<double>& x,
                     const std::vector<std::vector<double>>& matrix,
                     std::vector<double>& error) {
    double max_error = 0;
    for (std::size_t i = 0; i < x.size(); i++) {
        double xj = -matrix[i][matrix.size() - 1];
        for (std::size_t j = 0; j < x.size(); j++) {
            xj += matrix[i][j] * x[j];
        }
        xj -= matrix[i][i];
        xj /= -matrix[i][i];
        xj += x[i];
        error[i] = std::abs(x[i] - xj);
        if (error[i] > max_error) {
            max_error = error[i];
        }
        x[i] = xj;
    }
    return max_error;
}
```

```
result solve_iter_method(const std::vector<std::vector<double>>& matrix, double precision) {
    std::vector<double> x(n: matrix.size());
    std::vector<double> error(n: matrix.size());
    std::size_t count = 0;
    while (iterate(&x, matrix, &error) > precision) {
        count++;
    }
    return {count, x, error};
}
```

Примеры

Рандомизированная матрица небольшого размера:

```
Enter number of linear equations: 2

Enter accuracy epsilon: 0.001
Choose input type: "console", "random" or "file": random

Your matrix after conversions:
401 -21.1 56.6
59.7 530 -60.5

Time elapsed while solving system of equations: 2[μs]

Solution vector:
1.05 1.88

Error vector:
3.52e-06 3.96e-07

Number of iterations:
3
```

Рандомизированная матрица размером 20

```
Enter number of linear equations: 20

Enter accuracy epsilon: 0.001
Choose input type: "console", "random" or "file": random

Your matrix after conversions:
3112    -21.12  56.62  59.69  82.33   -68.49  -32.96  53.65   -44.45  10.79   -4.521  25.77   -27.04  2.68   98.45  83.24  27.14  43.46   -71.68  21.39   -96.74
-51.42  2574   68.84  -68.66  -19.81  -74.04  -78.24  99.78   -56.35  2.586  67.82  22.53   -40.79  27.51  4.857  -1.283  94.56  -41.5   54.27  5.349  53.98
-19.95  78.31  3836   -29.51  61.54  83.81   -86.05  89.87   5.199  -82.79  -61.56  32.65   78.05  -30.22  -87.17  -96   -8.46   -87.38  -52.34  94.13   80.44
70.18   -46.67  7.952  2711   52.05  2.507  33.54  6.321   -92.14  -12.47  86.37  86.16   44.19  -43.14  47.71  28   -29.19  37.57  -66.81  -11.98  76.02
65.84   -33.93  -84.21  78.67  2850  37.33  91.29  17.73  31.46  71.74   -12.09  84.79   -20.31  62.95  36.84  82.19  -3.502  -56.84  98.05  84.03  -70.47
76.21   28.22  -13.61  23.92  -43.79  3944   -38.51  -10.59  -54.78  -62.49  -44.75  11.29   -16.7   -66.08  81.36  -79.37  -74.78  -0.9112  52.1   96.95  87
36.89   -23.36  49.95  -26.27  -41.17  -53.55  2823   -51.12  -69.52  46.43   -74.91  58.69   -67.18  49.01  -85.09  98.02  -89.49  4.313  -64.76  -51.99  59.56
46.53  31.31  93.48  27.89  51.95  -81.3   -73.02  3811   -84.35  -86.02  -59.07  -7.716  63.94  14.66  51.12  -89.61  -68.44  100  -59.13  77.99  -74.91
99.56   -89.19  74.11  -85.53  -99.17  84.61  18.78   -63.93  3484  -21.66  82.61  63.94   -28.18  10.5   15.89  -9.485  37.48  -80.07  6.162  51.46  -39.14
98.45  15.39  75.52  49.56  25.78  -92.92  49.56  66.65  85.08  2176  66.21  95.89  48.76  80.67  96.72  33.38  -0.5483  -67.21  66  77.79  -84.6
29.94   -50.39  25.9   -54.17  40.12  -36.63  -34.24  -53.71  -85.17  26.61  2641  30.23  2.137  94.29  -43.99  9.221  43.85  -77.34  -5.703  18.51  88.86
-9.816  -32.73  69.54  -13.1   -99.35  -31.01  19.7   66.65  -53.22  35.1   -3.41  3734  -39.01  42.42  -63.49  24.36  -91.83  -17.2  39.2   34.79  27.53
-30.58  -63.08  21.82  25.43  46.15  -34.33  48.09  -59.56  84.18  36.95  30.63  -48.55  3415  -82.47  -47.9  75.48  37.22  -81.25  -77.74  -27.68  15.34
18.64  33.31  -42.24  55.15  -42.32  -34.07  -62.05  96.87  -99.28  65.48  -33.7   -62.36  -12.7  3856  83.79  52.97  39.82  -75.77  37.16  -23.23  54.85
88.61  83.25  72.38  -59.29  58.73  9.608  -40.54  80.99  81.93  74.8   -0.3712  15.24  -67.45  -45.22  2747  -1.52  -7.268  69.79  -0.8045  -41.79  -63.92
36.84  45.51  -72.19  20.62  -1.516  67.63  44.85  -64.36  -55.61  -0.2949  -75.75  -72.35  -27.91  -35.04  86.38  2564  24.42  67.37  63.63  -0.7851  -33.01
-21.13  31.77  21.78  -48.22  -69.75  -85.49  -78.43  29.44  -27.28  -42.35  -33.72  -81.77  -14.53  86.9  16.71  -46.91  2516  52.36  -2.515  -68.55  76.61
25.13  3.543  -58.43  11.51  -14.76  65.99  -21.12  -51.13  -34.8  45.87  27.73  96.97  -32.35  79.51  -72.79  -17.84  -98.92  3386  54.88  -41.26  -77.07
73.11  44.2   -98.17  -10.18  97.29  41.58  -57.82  -5.221  73.04  -81.22  -80.09  -23.42  -39.65  31.42  61.82  -73.66  -89.7  -89.32  2330  56.17  38.42
-11.49  -76.18  17.93  15.73  5.98  19.01  -27.62  -39.14  77.74  -4.683  -66.04  21.95  5.149  23.79  19.24  -53.27  65.96  -85.98  -80.23  3517  -66.07

Time elapsed while solving system of equations: 90[us]
```

```
Solution vector:
0.9188  0.9878  1.024   0.9302  0.7779  1.032   1.138   1.014   0.9673  0.6034  1.849   1.029   1.049   1.011   0.8858  0.9818  1.145   1.015   1.082   2.051

Error vector:
6.288e-06  2.467e-05  1.465e-05  5.899e-06  1.450e-05  7.69e-06  1.997e-06  8.700e-06  1.955e-06  3.173e-06  8.433e-07  7.419e-07  4.982e-07  1.72e-06  6.889e-07  2.128e-07  8.403e-07  9.584e-09

Number of iterations:
4
```

Матрица, считанная с файла:

Your matrix:

```
200 -4 2 5 7 -4 3 0 -4 -2 -8 -3 -2 8 -1 4 -2 1 6 -5 3
-2 200 1 -4 1 10 -3 9 -4 3 -7 -5 1 -9 -9 4 4 -4 8 8 1
-0 -5 200 -4 -1 -1 -1 -3 8 -8 3 -5 -2 8 -1 1 -0 3 2 6 1
4 1 -4 8 200 -4 4 -1 1 -9 -3 -5 3 0 -1 4 -10 9 -5 -9 -1
5 4 -10 -3 8 -2 -9 -3 6 200 -7 1 -10 5 6 -8 -2 -7 4 6 -9
3 8 1 -2 0 -7 221 -2 -8 1 5 -5 -9 -8 -2 -2 7 7 10 -7 6
8 6 2 3 3 5 -8 200 -6 -1 -3 -3 -1 -8 -5 9 -5 2 -6 -8 -8
9 4 -5 -1 9 364 2 -4 -0 3 -6 -2 3 9 -5 -5 7 -2 5 -6 -1
0 1 5 -8 9 3 3 5 432 4 3 8 -6 8 -2 -3 -1 8 0 -5 -2
-9 -5 0 -0 6 7 -0 -1 2 5 213 -2 -5 -2 -9 -2 -6 10 -5 -7 3
9 3 4 6 3 -2 -3 -9 2 8 8 321 7 -2 7 2 -3 7 -0 7 -9
-4 9 5 5 4 -9 0 8 7 -7 2 2 -8 142 -0 10 0 2 7 10 10
-10 -8 -7 0 -3 -9 -4 -10 -2 6 -5 -1 8 -4 200 10 -6 -1 -8 -8 -6
-1 9 2 202 -6 -5 -3 -1 6 -0 -8 4 3 4 -4 6 -2 5 2 1 2
-4 -4 -4 -9 -1 -5 1 -9 -6 3 5 4 2 8 -2 200 8 9 4 7 1
1 2 -5 -7 5 1 -4 -1 4 -3 6 -8 -9 -9 10 8 200 -4 7 -4 2
-4 -5 -5 -1 -1 1 -0 1 -0 1 -1 3 200 7 -1 5 -1 -7 6 -1 5
7 4 2 10 -6 0 -2 -6 1 -9 -2 -2 7 8 -3 -9 -7 -4 7 200 0
-7 4 5 -1 -4 5 10 -5 8 9 6 0 -10 8 -7 7 -4 200 -7 -4 4
1 8 -4 -3 7 7 -0 5 4 -5 6 -5 7 9 -6 -0 -8 -7 200 2 -1
```

Your matrix after conversions:

```
200 -4 2 5 7 -4 3 0 -4 -2 -8 -3 -2 8 -1 4 -2 1 6 -5 3
-2 200 1 -4 1 10 -3 9 -4 3 -7 -5 1 -9 -9 4 4 -4 8 8 1
-0 -5 200 -4 -1 -1 -1 -3 8 -8 3 -5 -2 8 -1 1 -0 3 2 6 1
-1 9 2 202 -6 -5 -3 -1 6 -0 -8 4 3 4 -4 6 -2 5 2 1 2
4 1 -4 8 200 -4 4 -1 1 -9 -3 -5 3 0 -1 4 -10 9 -5 -9 -1
9 4 -5 -1 9 364 2 -4 -0 3 -6 -2 3 9 -5 -5 7 -2 5 -6 -1
3 8 1 -2 0 -7 221 -2 -8 1 5 -5 -9 -8 -2 -2 7 7 10 -7 6
8 6 2 3 3 5 -8 200 -6 -1 -3 -3 -1 -8 -5 9 -5 2 -6 -8 -8
0 1 5 -8 9 3 3 5 432 4 3 8 -6 8 -2 -3 -1 8 0 -5 -2
5 4 -10 -3 8 -2 -9 -3 6 200 -7 1 -10 5 6 -8 -2 -7 4 6 -9
-9 -5 0 -0 6 7 -0 -1 2 5 213 -2 -5 -2 -9 -2 -6 10 -5 -7 3
9 3 4 6 3 -2 -3 -9 2 8 8 321 7 -2 7 2 -3 7 -0 7 -9
-4 -5 -5 -1 -1 1 -0 1 -0 1 -1 3 200 7 -1 5 -1 -7 6 -1 5
-4 9 5 5 4 -9 0 8 7 -7 2 2 -8 142 -0 10 0 2 7 10 10
-10 -8 -7 0 -3 -9 -4 -10 -2 6 -5 -1 8 -4 200 10 -6 -1 -8 -8 -6
-4 -4 -4 -9 -1 -5 1 -9 -6 3 5 4 2 8 -2 200 8 9 4 7 1
1 2 -5 -7 5 1 -4 -1 4 -3 6 -8 -9 -9 10 8 200 -4 7 -4 2
-7 4 5 -1 -4 5 10 -5 8 9 6 0 -10 8 -7 7 -4 200 -7 -4 4
1 8 -4 -3 7 7 -0 5 4 -5 6 -5 7 9 -6 -0 -8 -7 200 2 -1
7 4 2 10 -6 0 -2 -6 1 -9 -2 -2 7 8 -3 -9 -7 -4 7 200 0
```


Time elapsed while solving system of equations: 14[μs]

Solution vector:

1.011	0.9853	1.014	0.9681	1.091	0.9697	1.034	1.076	0.9328	1.084	1.117	0.8198	1.032	0.703	1.307	0.9799	1.017	0.9568
-------	--------	-------	--------	-------	--------	-------	-------	--------	-------	-------	--------	-------	-------	-------	--------	-------	--------

Error vector:

3.636e-06	2.32e-05	1.03e-05	6.111e-05	5.95e-06	7.533e-06	1.506e-06	3.595e-05	3.787e-06	9.738e-06	1.244e-05	1.945e-06
-----------	----------	----------	-----------	----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

Number of iterations:

4

Вывод

Алгоритмическая сложность алгоритма: $O(l * n^2)$, где l - число итераций, а n - размер матрицы.

Анализ применимости метода: Когда необходимо минимизировать погрешность вычислений, можно настраивать ϵ , чтобы добиться необходимой точности.

Вычисления можно производить параллельно и нет необходимости хранить всю матрицу в памяти.

Сравнение с другими методами:

- алгоритм Гаусса-Зейделя: скорость сходимости может быть больше, чем при прямых итерациях, но сложнее распараллеливать; также не обязательно держать всю матрицу в памяти;
- прямые методы: нужно держать матрицу в памяти; может накапливаться набегающая погрешность; если у СЛАУ есть решение, оно будет найдено, в отличие от итерационных методов, которые требуют условий сходимости; решение находится за конечное число операций;