Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет ИТМО»

Факультет Программной инженерии и компьютерной техники

Лабораторная работа №1
«Системы линейных алгебраических уравнений»
Вариант «Метод простых итераций»

Группа: P32312

Выполнили:
Воронин И.А.

Проверила:
Перл О.В.

Санкт-Петербург
2023

Описание метода

Метод простых итераций – итерационный метод решения систем линейных алгебраических уравнений. Суть метода заключается в том, чтобы последовательно уточнять начальное приближение к решению путем вычисления новых приближений на каждой итерации.

Условием сходимости процесса является наличие диагонального преобладания в матрице, т.е. :

$$|a_{ii}| \geqslant \sum_{j \neq i} |a_{ij}|,$$

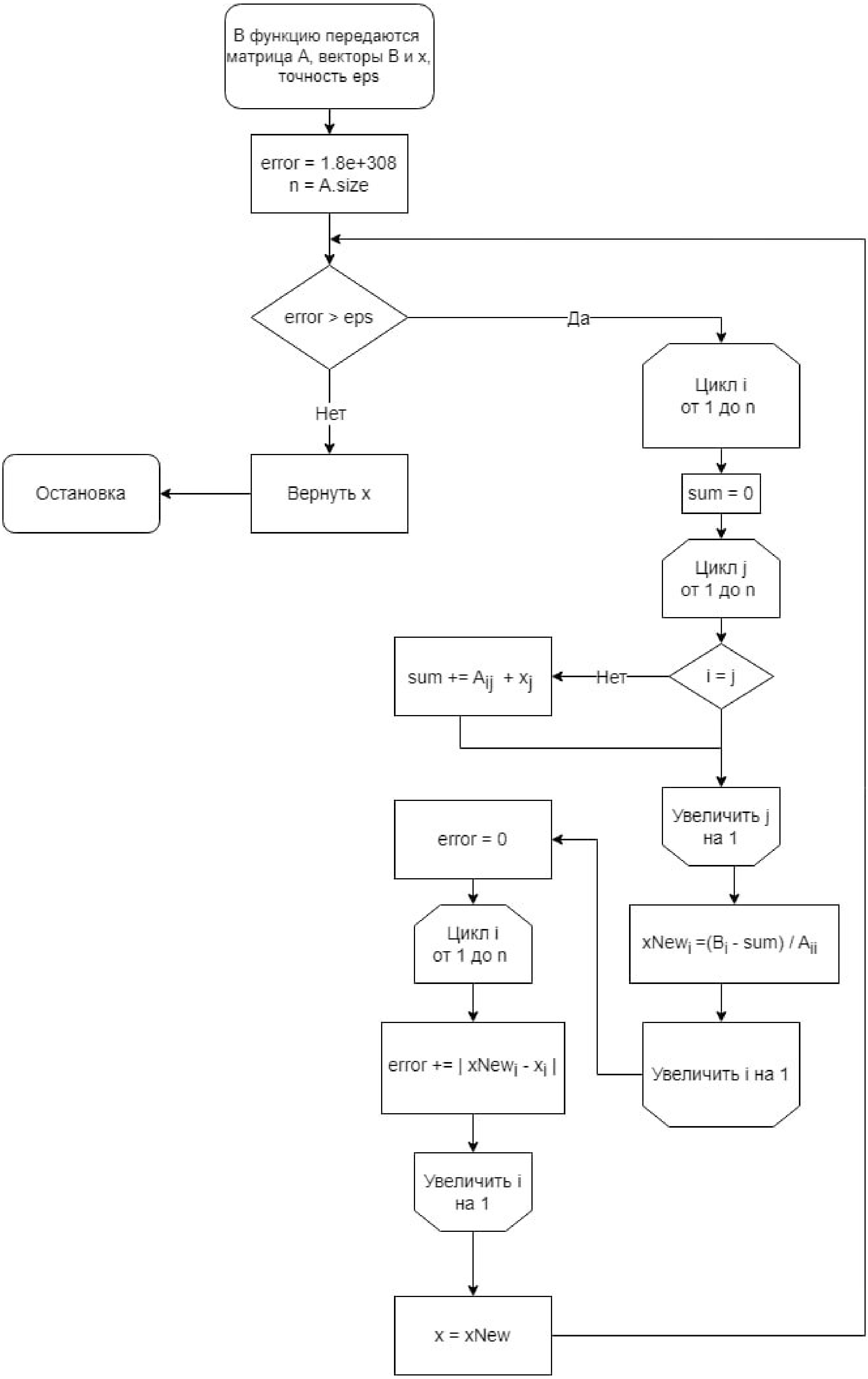
Расчетная формула:

$$x_i^{[j+1]} = x_i^{[j]} - \frac{1}{a_{ii}} f_i^{[j]} = x_i^{[j]} - \frac{1}{a_{ii}} \left(\sum_{k=1}^{n} a_{ik} x_k^{[j]} - b_i \right)$$

Условие окончания итерационного процесса:

$$\left| x_i^{[j]} - x_i^{[j-1]} \right| \leq \varepsilon$$

Блок-схема


В функцию передаются матрица A, векторы B и x, точность eps
error = 1.8e+308
n = A.size
error > eps
Да
Нет
Вернуть x
Остановка
Цикл i от 1 до n
sum = 0
Цикл j от 1 до n
i = j
Нет
sum += A_{ij} + x_j
Увеличить j на 1
$xNew_i$ =(B_i - sum) / A_{ii}
Увеличить i на 1
error = 0
Цикл i от 1 до n
error += | $xNew_i$ - x_i |
Увеличить i на 1
x = xNew

Листинг реализованного численного метода

```
fun solve(
    matrix: MutableList<DoubleArray>,
    variables: MutableList<Double>,
    x0: DoubleArray,
    eps: Double
): DoubleArray {
    val n = matrix.size
    var x = x0
    var iterationsCount = 0
    var error = Double.MAX_VALUE

    while (error > eps) {
        val xNew = DoubleArray(n) { 0.0 }

        for (i in 0 until n) {
            var sum = 0.0

            for (j in 0 until n) {
                if (i != j) {
                    sum += matrix[i][j] * x[j]
                }
            }

            xNew[i] = (variables[i] - sum) / matrix[i][i]
        }

        error = 0.0

        for (i in 0 until n) {
            error += abs(xNew[i] - x[i])
        }

        x = xNew
        iterationsCount++
    }

    println("Решение найдено за $iterationsCount итераций, погрешность:
$error")
    return x
}
```

Примеры работы

Ввод через консоль

```
console: solve
Введите размерность матрицы: 2
Введите коэффициенты матрицы через пробел
1 строка: 5 1
Введите свободный член: 2
2 строка: 4 13
Введите свободный член: 6
---Исходная матрица---

5.0 1.0 = 2.0
4.0 13.0 = 6.0

----------------------
Введите точность: 1e-9
Решение найдено за 16 итераций, погрешность: 7.198485141302058E-10

Приближения:
x1 = 0.32786885239158325
x2 = 0.36065573763074166
```

Ввод через файл

```
console: solve -f F:\code\comp-math-1\lab1\20.txt
---Исходная матрица---

96.532 -8.822 5.046 -0.466 -4.378 -0.656 8.17 -9.122 7.558 0.491 -7.101 -1.858 4.285 6.065 4.443 2.589 -7.426 5.664 -3.834 -7.558 = -736.63113
-2.358 101.002 -3.899 -2.048 -4.241 -6.544 4.306 -7.248 3.239 1.725 3.116 5.062 8.106 -0.025 6.727 -7.689 6.059 3.242 9.509 4.451 = -1011.40689
-0.647 9.114 92.375 2.049 -4.573 4.965 -1.183 1.807 9.593 4.805 -3.668 -1.432 -8.824 -2.749 6.382 -1.056 -8.417 9.863 -4.112 -6.136 = 595.03945
-5.681 8.765 -0.217 93.962 6.178 -3.91 7.717 0.278 -3.485 -7.578 -3.344 -2.009 -7.975 8.289 -4.318 -1.627 5.903 -9.668 -1.509 -4.511 = 749.0368
3.647 6.92 -3.217 -3.507 89.19 -7.276 -9.655 3.367 4.476 3.379 3.622 0.587 9.92 -4.966 -9.651 -2.055 8.968 -0.752 -1.125 1.1 = -11.09666
3.078 2.383 9.676 -9.423 9.152 121.852 0.7 -9.429 -4.32 9.531 7.397 8.374 -2.818 0.672 -8.155 -7.0 -3.412 9.248 -6.634 7.372 = -747.19795
2.038 -0.957 3.034 7.405 -7.896 3.256 100.872 -9.572 6.935 1.084 9.616 4.389 -0.585 -9.432 -5.463 -2.893 -4.084 7.778 3.763 -0.356 = 727.99241
-1.203 3.849 6.224 2.212 2.738 4.361 8.504 99.374 2.381 0.0 -4.492 3.728 -5.381 -8.96 -6.66 -3.358 -6.828 7.047 5.706 -4.632 = 373.30936
-1.255 -1.553 7.781 -6.014 7.767 0.104 7.308 5.893 110.686 -8.53 -0.396 -2.272 -5.885 -6.758 5.194 6.81 -4.808 -0.634 8.894 -3.172 = -1176.39383
-5.446 -8.916 4.107 -9.339 -5.204 9.528 -8.803 -9.535 -7.256 148.384 8.057 -5.724 2.652 -9.948 -4.598 8.443 -1.773 4.971 -5.743 -8.803 = 834.50263
1.71 -1.47 5.905 -3.999 -9.497 7.798 -1.854 -0.168 7.98 -2.319 88.685 8.54 -5.34 -5.809 -2.637 -6.326 -6.675 2.971 0.576 6.111 = -158.56869
7.927 -9.261 1.547 -6.548 1.463 -6.81 -0.208 8.367 6.965 4.395 4.637 106.722 -3.568 2.845 2.554 8.203 4.66 -5.707 -8.27 3.051 = 594.30402
-5.236 3.05 -8.755 -4.264 -2.008 5.861 8.039 4.769 -5.098 -8.0 3.681 -4.914 97.681 3.783 -2.209 -5.642 -7.451 -0.96 4.608 8.353 = -36.75088
-8.985 3.286 -7.825 2.908 8.228 1.346 4.809 -4.529 -1.335 -1.322 -1.154 -6.941 -5.206 80.152 -2.31 0.301 7.715 -6.529 -2.67 1.753 = 95.36961
2.063 2.395 -9.044 -9.454 2.808 9.062 -5.238 3.845 2.111 5.253 -2.878 4.698 9.169 -6.251 99.66 9.071 2.828 -0.236 -5.486 -6.77 = -883.73857
6.237 3.322 -5.165 -7.973 -1.17 -9.38 -9.898 4.057 -5.166 -4.431 -2.721 -3.4 -2.133 -1.048 -9.108 103.046 2.363 -4.58 2.285 0.047 = -236.03629
6.554 3.425 -8.275 -1.639 -4.375 -0.586 -2.074 1.231 -5.113 3.707 -7.185 -1.477 4.3 5.826 9.073 6.121 91.764 -6.022 8.442 -5.339 = -151.28169
-2.723 -0.316 9.843 -1.902 1.302 4.798 8.872 7.852 -9.534 -5.656 2.229 3.184 7.672 9.683 9.896 -6.89 3.666 113.989 1.206 2.175 = 1293.62312
9.947 1.862 -5.537 8.077 3.567 3.125 -8.181 -4.902 3.844 -8.558 3.843 -4.614 1.839 -0.724 -6.259 -8.118 -9.737 4.323 99.002 -0.945 = -932.39781
-5.579 -0.637 0.586 5.339 6.628 6.143 9.177 3.011 -5.147 -9.226 -2.764 2.485 6.962 -2.67 7.822 5.38 0.021 7.735 -8.328 96.64 = 554.26341

----------------------
Введите точность: 1e-5
Решение найдено за 14 итераций, погрешность: 6.485985061477351E-6

Приближения:
x1 = -8.752999984567358
x2 = -9.92900004684128
x3 = 8.046999872659741
x4 = 8.126000207457595
x5 = 0.9439999840254698
x6 = -8.470000068332652
x7 = 6.910999845251473
x8 = 2.7949999312587903
x9 = -9.857000084801886
x10 = 5.482999920911698
x11 = -0.9480001232371623
x12 = 5.428000027579405
x13 = 0.6339999445082308
x14 = 1.4650001013457468
x15 = -6.4020000791778635
x16 = -0.701999842034235
x17 = 1.6260001047003714
x18 = 9.78799994465791
x19 = -7.1740000767541305
x20 = 3.3789998657206937
```

Случайно сгенерированная матрица

```
console: solve -r 5
---Исходная матрица---

304.9840461270921 32.825023380820085 -81.36653452681016 -89.96577101374996 -59.97880182654567 = 87.60695708642822
81.09622667803882 411.60536556969976 -77.72382783172331 13.101551818661179 89.01291699860607 = 62.18731854502337
48.41818386146147 -84.67331646132399 448.084101276065 -32.03155783536327 -86.897245799304 = 75.08068254615588
-87.4740145404221 -7.6757298652097035 -56.47919195947886 364.69817959716886 -83.00677644093884 = 12.088106593834596
-58.19790175839123 -71.47120603502164 9.711464264198185 -60.82846840868334 239.015992695225 = -62.72969240380102

----------------------
Введите точность: 1e-7
Решение найдено за 18 итераций, погрешность: 5.6665239028808045E-8

Приближения:
x1 = 0.3179993906970566
x2 = 0.13853354438750176
x3 = 0.14329263041611462
x4 = 0.10669777177803158
x5 = -0.12226359459231047
```

Вывод

Если обозначить число итераций за *l*, тогда алгоритмическая сложность метода простой итерации O(*l* * *n*²) т.к. не известно количество итераций, которое может быть больше или меньше n.

Главные отличия метода простой итерации от метода Гаусса-Зейделя:

1) Условие сходимости метода Гаусса-Зейделя более строгое, из-за чего сложнее найти систему для применения метода.
2) Скорость сходимости метода Гаусса-Зейделя может быть быстрее, т.к. используются свежеполученные значения неизвестных с текущей итерации.
3) Метод Гаусса-Зейделя сложнее параллелизовать т.к. все уравнения зависят друг от друга.

Прямые методы позволяют найти точное решение системы за определенное количество шагов, но плохо работают на больших наборах данных в отличии от итерационных. Но из-за строгих условий итерационные методы применимы в меньшем количестве случаев.