Федеральное государственное автономное образовательное учреждение высшего образования **«Национальный исследовательский университет ИТМО»**

Факультет Программной Инженерии и Компьютерной Техники

**Лабораторная работа №1**

**по дисциплине «Вычислительная математика»**

Вариант: **4**

Преподаватель:   
Наумова Надежда Александровна

Выполнил: Кирячек Тимофей

Группа: Р3209

Санкт-Петербург, 2025 г

# Цель работы

Изучить численные методы решения систем линейных алгебраических уравнений и реализовать один из них средствами программирования.

# Описание метода

Итерационные методы дают возможность для системы (1) построить последовательность векторов 𝑥 (0) , 𝑥 (1) , … , 𝑥 (𝑘) , пределом которой должно быть точное решение 𝑥 (∗) : 𝑥 (∗) = lim 𝑘→∞ 𝑥 (𝑘) Построение последовательности заканчивается, как только достигается желаемая точность.

Приведем систему уравнений, выразив неизвестные 𝑥1, 𝑥2, … , 𝑥𝑛 соответственно из первого, второго и т.д. уравнений системы.

# Листинг программы

<https://github.com/TecHeReTiC3141/CalcMaths_Lab1/blob/master/src/utils.ts>

## utils.ts

*const* hasDiagonalDominance = (A: *number*[][]): *boolean* => {  
 *return* A.every((row, i) => {  
 *const* diag = Math.abs(row[ i ]);  
 *const* sum = row.reduce((acc, val) => acc + Math.abs(val), 0);  
 *return* sum - diag <= diag;  
 })  
}  
  
*const* enforceDiagonalDominance = (A: *number*[][], b: *number*[]): *void* => {  
 *const* n = A.length;  
 *for* (*let* i = 0; i < n; ++i) {  
 *const* max = A[ i ].reduce((acc, cur) => Math.max(acc, cur));  
 *const* maxIndex = A[ i ].findIndex((val) => val === max);  
 *if* (maxIndex !== i) {  
 [ A[ i ], A[ maxIndex ] ] = [ A[ maxIndex ], A[ i ] ];  
 [ b[ i ], b[ maxIndex ] ] = [ b[ maxIndex ], b[ i ] ];  
 }  
 }  
}  
  
*const* MAX\_ITERATIONS = 2000  
  
*const* shouldStopIteration = (x: *number*[], xPrev: *number*[], eps: *number*): *boolean* => {  
 *return* x.every((val, i) => Math.abs(val - xPrev[ i ]) < eps);  
}  
  
*type Result* = [*number*[], *number*, *number*[]]  
  
*const* simpleIterationMethod = (A: *number*[][], b: *number*[], x0: *number*[], eps: *number*): *Result* => {  
 *const* C = A.map((row, i) => row.map((val, j) => i === j ? 0 : -val / A[ i ][ i ]));  
 *const* d = b.map((val, i) => val / A[ i ][ i ]);  
 *const* x = [...d];  
 *let* xPrev = [];  
  
 *const* n = A.length;  
 *for* (*let* iter = 0; iter < MAX\_ITERATIONS; ++iter) {  
 xPrev = [...x]  
 *for* (*let* i = 0; i < n; ++i) {  
 x[i] = d[i] + C[i].reduce((acc, val, j) => acc + val \* xPrev[j], 0);  
 }  
 *if* (shouldStopIteration(x, xPrev, eps)) {  
 *const* deviations = x.map((val, i) => Math.abs(val - xPrev[i]))  
 *return* [x, iter, deviations]  
 }  
 }  
 *return* [[], -1, []];  
}  
  
*const calculateEquationSolution* = (coeffs: *string*[][], eps: *number*): *Result* => {  
 *const* A = coeffs.map(row => row.slice(0, -1).map(val => +val));  
 *const* b = coeffs.map(row => +row[row.length - 1]);  
 *if* (!hasDiagonalDominance(A)) enforceDiagonalDominance(A, b);  
 *return* simpleIterationMethod(A, b, *new* Array(b.length).fill(0), eps);  
}  
  
*export* { *calculateEquationSolution* }

# Примеры работы программы

Введите:

1 - Ввести матрицу с консоли

2 - Ввести матрицу с файла

3 - Генерация случайной матрицы

1

Вводим матрицу с консоли

Введите размерность матрицы:

3

Введите строки матрицы:

10 2 -1 5

-2 -6 -1 24.4

1 -3 12 36

Введите точность:

0.01

Матрица:

10 2 -1 5

-2 -6 -1 24.4

1 -3 12 36

№ |x1 |x2 |x3 | eps1 | eps2 | eps3 |

0 |0 |0 |0 |

1 |0,500000 |-4,236667 |1,899167 | 0,500000 |4,236667 |1,899167 |

2 |1,537250 |-4,898944 |1,647160 | 1,037250 |0,662278 |0,252007 |

3 |1,644505 |-4,892695 |1,639784 | 0,107255 |0,006250 |0,007376 |

4 |1,642517 |-4,890803 |1,640423 | 0,001987 |0,001892 |0,000639 |

Решение системы:

[1] = 1,642517401620371

[2] = -4,890803167438272

[3] = 1,640422758005401

Вектор невязки:

[1] = 0,003144923321760

[2] = -0,000638556616508

[3] = 0,000000000000000

# Вывод:

В результате выполнения данной лабораторной работой я познакомился с численными методами решения математических задач на примере систем алгебраических уравнений, реализовав веб-приложение для решения СЛАУ методом простых итераций.