

Федеральное государственное автономное  
образовательное учреждение высшего образования  
Национальный Исследовательский Университет ИТМО

## Лабораторная работа №1

Дисциплина: Вычислительная математика  
Вариант №6

Выполнил: Марков Кирилл Андреевич  
Факультет: Программной инженерии и компьютерной техники  
Группа: P3213  
Преподаватель: Машина Екатерина Алексеевна

Город Санкт-Петербург  
2024 год

## Цель работы

Реализовать программу для решения систем линейных алгебраических уравнений с использованием метода простых итераций. Данная программа должна быть способна принимать на ввод коэффициенты матрицы как с клавиатуры, так и из файла. Также, программа должна выполнять ряд функций, включая вычисление вектора неизвестных, проверку диагонального преобладания, вывод количества итераций, за которое было найдено решение, а также вывод вектора погрешностей.

## Описание метода, расчетные формулы

Рабочая формула метода простой итерации:

$$x_i^{(k+1)} = \frac{b_i}{a_{ii}} - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{a_{ij}}{a_{ii}} x_j^k, \quad i = 1, 2, \dots, n$$

выполнение условия **преобладания диагональных элементов** или доминирование диагонали:

$$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|, \quad i = 1, 2, \dots, n$$

## Листинг программы

```
print(f"Matrix:\n{matrix}\n")
print(f"Results:\n{results}\n")
if matrix.get_max_by_module() == 0:
    print("Matrix max element is 0 :(")
    return

matrix.rearrange(results)
print(f"Matrix after rearrange:\n{matrix}\n")
print(f"Results after rearrange:\n{results}\n")

right_side_matrix: SquareMatrix = get_right_side_matrix(matrix)
print(f"Matrix after division:\n{right_side_matrix}\n")

right_side_results: Matrix = get_right_side_results(matrix, results)
print(f"Results after division:\n{right_side_results}\n")

x: Matrix = right_side_results
new_x: Matrix = right_side_results
diff: float = float("inf")
iteration_count: int = 0
while diff > precision:
    iteration_count += 1
    x = new_x
    new_x = right_side_matrix * x + right_side_results
```

```

    diff = (new_x - x).get_max_by_module()
print("Error vectors: ")
print(new_x - x)
print()

print(f"To get result script needed {iteration_count} iterations")
print()

print("Solution:")
print(new_x)
print()

```

Welcome!

How would you like to input data? (with cli - 1, from file - 2): 1

Please input matrix dimension

(make sure it's lower than 21 and greater than 0): 2

Please input precision

(make sure it's greater than 1e-9): 0.000000001

Would you like to input matrix by yourself? y/n n

Matrix:

[65, -39]

[-18, 19]

Results:

[-38]

[18]

Matrix after rearrange:

[65, -39]

[-18, 19]

Results after rearrange:

[-38]

[18]

Matrix after division:

[0, 0.6]

[0.9473684210526315, 0]

Results after division:

[-0.5846153846153846]

[0.9473684210526315]

Error vectors:

[8.371353610314713e-10]

[-8.15670420095671e-10]

To get result script needed 73 iterations

Solution:

[-0.03752345212618646]

[0.9118198866663266]

Вывод

Цель работы была достигнута - разработана и реализована эффективная программа для решения систем линейных уравнений методом простых итераций. В результате работы программы были получены точные решения с учетом заданной точности, что подтверждается выводом вектора неизвестных.