

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: **6**

Преподаватель:
Машина Екатерина Алексеевна

Выполнил: Молодиченко Семен
Группа: P3213

Санкт-Петербург, 2025 г

Цель работы

Изучить методы численного решения систем линейных алгебраических уравнений и реализовать один из них с помощью программирования.

Описание метода

Метод Гаусса-Зейделя – это численный итерационный метод решения системы линейных алгебраических уравнений (СЛАУ). Данный метод является модификацией метода простых итераций, где при вычислении каждого элемента используются уже найденные на текущем шаге значения.

Формулы метода: Для системы уравнений:

$$Ax=b$$

итерационная формула имеет вид:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} \right)$$

Метод сходится, если матрица обладает диагональным преобладанием или является положительно определенной.

Листинг программы

<https://github.com/semchik200001/mathematics-/blob/main/main.py>

```
import numpy as np

def read_matrix_from_file(filename):
    """Считывает матрицу A и вектор b из файла."""
    with open(filename, 'r') as file:
        while True:
            try:
                n = int(file.readline())
                if n > 20:
                    print("ValueError: Размерность матрицы должна быть не
более 20.")
                    continue
                break
            except ValueError:
                print("ValueError: Размерность матрицы должна быть не более
20.")
        A = [list(map(float, file.readline().split())) for _ in range(n)]
        b = list(map(float, file.readline().split()))
        return np.array(A), np.array(b)

def read_matrix_from_input():
    """Считывает матрицу A и вектор b с клавиатуры."""
    while True:
        try:
            n = int(input("Введите размерность матрицы (не более 20): "))
```

```

        if n > 20:
            print("ValueError: Размерность матрицы должна быть не более
20.")
            continue
        break
    except ValueError:
        print("ValueError: Размерность матрицы должна быть не более 20.")
A = []
print("Введите коэффициенты матрицы A (по строкам, через пробел):")
for _ in range(n):
    while True:
        try:
            row = list(map(float, input().split()))
            if len(row) != n:
                raise ValueError(f"Ожидалось {n} чисел, введено
{len(row)}.")
            A.append(row)
            break
        except ValueError as e:
            print("Ошибка ввода! Повторите ввод строки:", e)
print("Введите коэффициенты вектора b (через пробел):")
while True:
    try:
        b = list(map(float, input().split()))
        if len(b) != n:
            raise ValueError(f"Ожидалось {n} чисел, введено {len(b)}.")
        break
    except ValueError as e:
        print("Ошибка ввода! Повторите ввод:", e)
return np.array(A), np.array(b)

def check_diagonal_dominance(A):
    """Проверяет, обладает ли матрица диагональным преобладанием."""
    n = A.shape[0]
    for i in range(n):
        sum_row = sum(abs(A[i, j]) for j in range(n) if i != j)
        if abs(A[i, i]) < sum_row:
            return False
    return True

def enforce_diagonal_dominance(A, b):
    """Пытается добиться диагонального преобладания путем перестановки
строк."""
    n = A.shape[0]
    indices = np.argsort(-np.abs(A.diagonal()))
    A, b = A[indices], b[indices]
    if not check_diagonal_dominance(A):
        print("Невозможно достичь диагонального преобладания.")
        return None, None
    return A, b

def compute_determinant(A):
    """Вычисляет определитель матрицы A."""
    return np.linalg.det(A)

def gauss_seidel(A, b, tol=1e-6, max_iterations=1000):
    """Решает систему методом Гаусса-Зейделя."""
    n = len(A)
    x = np.zeros(n)
    for iteration in range(max_iterations):

```

```

        x_new = np.copy(x)
        for i in range(n):
            sum1 = sum(A[i][j] * x_new[j] for j in range(i))
            sum2 = sum(A[i][j] * x[j] for j in range(i + 1, n))
            x_new[i] = (b[i] - sum1 - sum2) / A[i][i]
        error = np.linalg.norm(x_new - x, ord=np.inf)
        if error < tol:
            return x_new, iteration + 1
        x = x_new
    print("Метод не сошелся за", max_iterations, "итераций.")
    return x, max_iterations

def main():
    choice = input("Вы хотите ввести данные с клавиатуры (k) или из файла (f)? ")
    if choice.lower() == 'f':
        filename = input("Введите имя файла: ")
        A, b = read_matrix_from_file(filename)
    else:
        A, b = read_matrix_from_input()

    print("Определитель матрицы:", compute_determinant(A))

    if not check_diagonal_dominance(A):
        print("Матрица не обладает диагональным преобладанием. Пытаемся переставить строки...")
        A, b = enforce_diagonal_dominance(A, b)
        if A is None:
            return

    print("Решаем методом Гаусса-Зейделя...")
    solution, iterations = gauss_seidel(A, b)
    print("Вектор неизвестных:", solution)
    print("Количество итераций:", iterations)

    # Вычисление невязки
    residual = b - np.dot(A, solution)
    print("Вектор невязок:", residual)

    # Вычисление нормы погрешности
    print("Норма погрешности:", np.linalg.norm(residual))

    # Проверка с решением с помощью numpy.linalg.solve
    lib_solution = np.linalg.solve(A, b)
    print("Решение с использованием библиотеки:", lib_solution)
    print("Разница между решениями:", np.linalg.norm(solution - lib_solution))

if __name__ == "__main__":
    main()

```

Примеры и результаты работы

Входные данные:

4 1 2

3 5 1

1 1 3

4 7 3

Результат выполнения программы:

Определитель матрицы: 44.000000000000014

Решаем методом Гаусса-Зейделя...

Вектор неизвестных: [0.50000005 0.99999998 0.49999999]

Количество итераций: 11

Вектор невязок: [-1.63840000e-07 -4.09599998e-08 0.00000000e+00]

Норма погрешности: 1.6888240661213822e-07

Решение с использованием библиотеки: [0.5 1. 0.5]

Разница между решениями: 5.608678994819455e-08

Выводы

- Метод Гаусса-Зейделя успешно применяется для решения СЛАУ при выполнении условия сходимости (диагональное преобладание);
- Результаты, полученные методом Гаусса-Зейделя, совпадают с решением, найденным библиотечными методами;
- Метод требует меньшего количества итераций по сравнению с методом простых итераций;
- Программа корректно выполняет проверку условий сходимости и обрабатывает случаи, когда диагональное преобладание отсутствует.