

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 №21

*Выполнил:*  
Студент группы Р3213  
Хафизов Булат Ленарович  
*Преподаватель:*  
Мальшева Татьяна  
Алексеевна

## Цель работы

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

## Описание метода

Метод Гаусса.

Основан на приведении матрицы системы к треугольному виду так, чтобы ниже ее главной диагонали находились только нулевые элементы.

**Прямой ход** метода Гаусса состоит в последовательном исключении неизвестных из уравнений системы. Сначала с помощью первого уравнения исключается  $x_1$  из всех последующих уравнений системы. Затем с помощью второго уравнения исключается  $x_2$  из третьего и всех последующих уравнений и т.д. Этот процесс продолжается до тех пор, пока в левой части последнего ( $n$ -го) уравнения не останется лишь один член с неизвестным  $x_n$ , т. е. матрица системы будет приведена к треугольному виду.

**Обратный ход** метода Гаусса состоит в последовательном вычислении искомых неизвестных: решая последнее уравнение, находим единственное в этом уравнении неизвестное  $x_n$ . Далее, используя это значение, из предыдущего уравнения вычисляем  $x_{n-1}$  и т. д. Последним найдем  $x_1$  из первого уравнения. Метод имеет много различных вычислительных схем, но в каждой из них основным требованием является  $\det A \neq 0$ .

## Листинг программы

```
class Calculator:
    n = 0
    system = []
    x = []

    def __init__(self, n, system):
        self.n = n
        self.system = system
        self.x = []

    def calculate(self):
        try:
            print("Исходная система:")
            self.print_system()
            self.triangle()
            print("Треугольная система:")
            self.print_system()
            self.solution()
            self.print_x()
            self.get_residuals()
        except ArithmeticError:
            return

    def print_x(self):
        print("Решение:")
        for i in range(self.n):
            print("\tX_" + str(i + 1) + " = " + str(self.x[self.n - i - 1]))

    def solution(self):
        self.x.append(self.system[self.n - 1][-1] / self.system[self.n - 1][self.n - 1])
        for i in range(self.n - 2, -1, -1):
```

```

        value = self.system[i][-1]
        k = self.n - 1
        while k > i:
            value -= self.x[self.n - 1 - k] * self.system[i][k]
            k -= 1
        self.x.append(value / self.system[i][i])

    def check_diagonal(self, i):
        j = i
        while j < self.n:
            if self.system[j][i] != 0 and self.system[i][j] != 0:
                swap = self.system[j]
                self.system[j] = self.system[i]
                self.system[i] = swap
                return
            j += 1
        print("Нет решений!")
        return ArithmeticError

    def print_system(self):
        for i in range(self.n):
            for j in range(self.n):
                print(str(self.system[i][j]) + " x_" + str(j + 1) + " ",
end='')
            print(str(self.system[i][-2]) + " " + str(self.system[i][-1]),
end='')
            print("")

    def triangle(self):
        try:
            for i in range(self.n):
                if self.system[i][i] == 0:
                    self.check_diagonal(i)
                m = i
                while m < self.n - 1:
                    a = -(self.system[m + 1][i] / self.system[i][i])
                    j = i
                    while j < self.n:
                        self.system[m + 1][j] += a * self.system[i][j]
                        j += 1
                    self.system[m + 1][-1] += a * self.system[i][-1]
                    m += 1
                k, check = 0, False
                while k < self.n:
                    if self.system[i][k] != 0:
                        check = True
                        break
                    k += 1
                if not check:
                    print("Нет решений!")
                    return ArithmeticError
        except ValueError:
            print("Неправильные данные!")
            return

    def get_residuals(self):
        i = 0
        print("Вектора невязок:")
        self.x.reverse()
        while i < self.n:
            res = 0
            j = 0
            while j < self.n:
                res += self.system[i][j] * self.x[j]

```

```

        j += 1
        res -= self.system[i][-1]
        i += 1
        print("\tВектор невязок для №" + str(i) + " = " + str(abs(res)))
    print("")

```

```

from lab1.Calculator import Calculator

def input_console():
    print("Введите размерность матрицы 1 < n <= 20:")
    n = int(input().strip())
    if 1 < n <= 20:
        print("Введите уравнения в формате:\n"
              "a11 a12 ... a1j | b1")
        a = []
        for i in range(n):
            while True:
                row = input().split()
                if len(row) - 2 == n and row[-2] == "|":
                    a.append(row)
                    break
            else:
                print("Проверьте правильность ввода! Попробуйте еще раз!")
        for i in range(len(a)):
            for j in range(len(a[i])):
                if a[i][j] != "|":
                    a[i][j] = float(a[i][j])
        det = determinant(a)
        if det == 0:
            print("Не имеет решений! Определитель равен 0!")
            return
        print("Детерминант: " + str(det))
        calculator = Calculator(n, a)
        calculator.calculate()
        del calculator
    else:
        print("Неправильное значение!")
        return

def input_file(path):
    n = 0
    a = []
    file = open(path, 'r', encoding='utf-8')
    for line in file:
        if line != "\n" and line != " " and line != " \n":
            n += 1
    file.close()
    file = open(path, 'r', encoding='utf-8')
    for row in file:
        line = row.split()
        if line[-2] == "|" and len(line) - 2 == n:
            a.append(line)
        else:
            print("Неправильное содержимое файла!")
    file.close()
    for i in range(len(a)):
        for j in range(len(a[i])):
            if a[i][j] != "|":
                a[i][j] = float(a[i][j])

```

```

det = determinant(a)
if det == 0:
    print("Не имеет решений! Определитель равен 0!")
    return
print("Детерминант: " + str(det))
calculator = Calculator(n, a)
calculator.calculate()
del calculator

def minor(matrix, i, j):
    n = len(matrix)
    return [[matrix[row][col] for col in range(n) if col != j] for row in
range(n) if row != i]

def determinant(matrix):
    n = len(matrix)
    if n == 1:
        return matrix[0][0]
    det = 0
    sgn = 1
    for j in range(n):
        det += sgn * matrix[0][j] * determinant(minor(matrix, 0, j))
        sgn *= -1
    return det

def main():
    print("Лабораторная работа #1\n"
          "Метод Гаусса\n"
          "Выберите доступную функциональность:\n"
          "\t1. Ввод с консоли.\n"
          "\t2. Чтение из файла.")
    action_type = input().strip()
    if action_type == "1":
        input_console()
    elif action_type == "2":
        input_file(input("Пропишите путь к файлу:\n").strip())
    else:
        print("Неверный ввод!")

main()

```

## Примеры работы программы

```

Лабораторная работа #1
Метод Гаусса
Выберите доступную функциональность:
    1. Ввод с консоли.
    2. Чтение из файла.
1
Введите размерность матрицы 1 < n <= 20:
3
Введите уравнения в формате:
a11 a12 ... a1j | b1
3 2 -5 1 -1
2 -1 3 1 -13
1 3 -4 1 9
Детерминант: -30.0
Исходная система:
3.0 x_1 2.0 x_2 -5.0 x_3 | -1.0
2.0 x_1 -1.0 x_2 3.0 x_3 | 13.0
1.0 x_1 2.0 x_2 -1.0 x_3 | 9.0
Треугольная система:
3.0 x_1 2.0 x_2 -5.0 x_3 | -1.0
0.0 x_1 -2.3333333333333333 x_2 6.333333333333333 x_3 | 13.666666666666666
0.0 x_1 -2.220446049250313e-16 x_2 4.2857142857142865 x_3 | 17.142857142857146
Решение:
X_1 = 3.0
X_2 = 5.0
X_3 = 4.0
Вектора невязок:
Вектор невязок для №1 = 0.0
Вектор невязок для №2 = 1.7763568394002505e-15
Вектор невязок для №3 = 0.0

```

## Вывод

В результате выполнения данной лабораторной работой я познакомился с численными методами решения математических задач на примере систем алгебраических уравнений, реализовав на языке программирования Python метод Гаусса.