



Федеральное государственное автономное образовательное  
учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет ПИ и КТ

BT

CompMath edition

## Лабораторная работа №1

по дисциплине: «Вычислительная математика»  
«Решение системы линейных алгебраических уравнений»  
Вариант 1: методом Гаусса

Выполнил:

**Болорболд Аригуун,**

группа Р3211

Преподаватель:

**Малышева Татьяна Алексеевна**

Санкт-Петербург

2024



## 1. Цель работы:

Научиться искать решение СЛАУ при помощи численных методов, которая будет совершать приближенные вычисления и находить решение, получая из вход матрицу из файла или консоли.

## 2. Задание лабораторной работы:

1. № варианта определяется как номер в списке группы согласно ИСУ. (1)
2. В программе численный метод должен быть реализован в виде отдельной подпрограммы или класса, в который входные/выходные данные передаются в качестве параметров.
3. Размерность матрицы  $n \leq 20$  (задаётся из файла или с клавиатуры – по выбору конечного пользователя).
4. Должна быть реализована возможность ввода коэффициентов матрицы как с клавиатуры, так и из файла (по выбору конечного пользователя).

Для метода Гаусса должно быть реализовано:

- ★ Точность задаётся с клавиатуры/файла;
- ★ Вывод треугольной матрицы (включая преобразованный вектор правых частей системы);
- ★ Вычисление осуществляется по треугольной матрице;
- ★ Вывод вектор неизвестных:  $x_1, x_2, \dots, x_n$ ;
- ★ Вывод вектора невязок:  $r_1, r, \dots, r_n$ ;
- ★ Используя библиотеки выбранного языка программирования найти решение системы линейных уравнений и значение детерминанта. Сравнить результаты;
- ★ Рассмотреть случаи, когда применение метода Гаусса невозможно.

## 3. Описание метода, расчётные формулы:

Метод Гаусса – прямой метод, находящий решение путём элементарных преобразования расширенной матрицы системы и приведения в треугольный вид так, чтобы ниже её диагонали находились только нулевые элементы (отсюда и другое название этого метода –

последовательного исключения неизвестных), откуда можно получать решение СЛАУ. Здесь требуются не знание формул, а только знание арифметических действий.

#### 4. Исходный код (*Python*):

```
from random import randint
from decimal import Decimal
import numpy as np
import math
def get_random_matrix(sz):
    mn = -1000
    mx = 1000
    an = [
        [randint(mn, mx) for _ in range(sz)]
        for _ in range(sz)
    ]
    bn = [randint(mn, mx) for _ in range(sz)]
    return an, bn
def print_line():
    print("\033[0;32m" + "=" * 50 + "=" + "\033[0m")
def print_condition(str):
    print("\033[0;36m" + str + "\033[0m")
def print_message(str):
    print("\033[0;31m" + str + "\033[0m")
def print_result(str):
    print("\033[0;35m" + str + "\033[0m")
def get_data():
    global b, cur
    random_word = 'RANDOM'
    random_word_russian = 'РАНДОМ'
    sn = input("\033[0;36m" + 'Введите размерность матрицы (n): ' +
"\033[0m").strip()
    while not sn.isdigit() or int(sn) < 1 or int(sn) > 20:
        print_line()
        print_message('Размер n должно быть числом на отрезке [1;20]')
        sn = input("\033[0;36m" + 'Введите размерность матрицы (' + sn +
'): ' + "\033[0m").strip()
    n_sz = int(sn)
    rw = input("\033[0;36m" + f'Введите "{random_word}" |
"{random_word_russian}" для генерации случайной матрицы (A и B): ' +
"\033[0m").strip()
    if rw.casefold() == 'RANDOM'.casefold() or rw.casefold() ==
'РАНДОМ'.casefold() or rw.casefold() == 'HFYLVJ'.casefold() or
rw.casefold() == 'КФТВЩЬ'.casefold():
        a, b = get_random_matrix(n_sz)
        print_result("\033[1;35m" + 'Сгенерированная матрица: ')
        for i in range(n_sz):
            print(*a[i], '|', b[i], sep='\t')
        print_line()
        print("\033[0m", end="")
    else:
        print_condition('Введите элементы матрицы A:')
        a = []
        for i in range(n_sz):
            fl = False
            while not fl:
                try:
```

```

        print_condition(f'Введите строку ({i}) (n чисел): ')
        cur = list(map(float, input().split()))
        assert len(cur) == n_sz
        fl = True
    except Exception:
        pass
    a.append(cur)

print_condition('Введите элементы матрицы B:')
fl = False
while not fl:
    try:
        print_condition('Введите строку (n чисел): ')
        b = list(map(float, input().split()))
        assert len(b) == n_sz
        fl = True
    except Exception:
        pass

return n_sz, a, b

def get_data_from_file():
    with open('input.txt') as f:
        n = int(f.readline())
        f.readline()
        a = [
            list(map(float, f.readline().split()))
            for _ in range(n)
        ]
        f.readline()
        b = list(map(float, f.readline().split()))
        return n, a, b

def get_determinant(triangular_matrix, k):
    res = 1
    for i in range(len(triangular_matrix)):
        res *= triangular_matrix[i][i]
    return (-1) ** k * res

def get_solution_and_k(a, b):
    x = [None] * n
    k_ = 0

    for i in range(0, n - 1):
        while a[i][i] == 0:
            a = a[:i] + a[i + 1:] + [a[i]]
            b = b[:i] + b[i + 1:] + [b[i]]
            k_ += 1
            if k_ > math.factorial(n):
                print_message("Матрица СЛАУ не соответствует теореме
Кронекера-Капелли.")
                exit(0)

        for k in range(i + 1, n):
            c = a[k][i] / a[i][i]
            a[k][i] = 0
            for j in range(i + 1, n):
                a[k][j] = a[k][j] - c * a[i][j]
            b[k] = b[k] - c * b[i]

    for i in range(n - 1, -1, -1):

```

```

        s = 0
        for j in range(i + 1, n):
            s = s + a[i][j] * x[j]
        if(not np.any(a)) == 0:
            print_message("Нет решения: несовместная система.")
            print_result(str(k))
            exit(0)
        x[i] = (b[i] - s) / a[i][i]
        print(a[i][i])

    return x, k_, a, b

def get_r(a, b, x):
    n = len(a)
    r = [0] * n
    for i in range(n):
        for j in range(n):
            r[i] += a[i][j] * x[j]
        r[i] -= b[i]
    return r

print("\033[1;33m" + "Вычислительная математика - Лабораторная №1" +
"\033[0m")
print_line()
print("Каким образом будет осуществлен ввод СЛАУ?")
while True:
    inp = input("\033[1;33m" + "Вручную (+)" + "\033[0m" + " | " +
"\033[0;35m" + "С файла \'input.txt\' (-): " + "\033[0m")
    if inp == "+":
        n, a, b = get_data()
        break
    elif inp == "-":
        n, a, b = get_data_from_file()
        break
    else:
        print_message("Ответ не распознан. Попробуйте снова.")

det = Decimal(str(np.linalg.det(np.array(a))))

initial_a = [el[:] for el in a]
initial_b = b[:]

x, k, a, b = get_solution_and_k(a, b)
if det == 0:
    print_message('Матрица СЛАУ является вырожденным, det A = 0.')
    print_result(str(k))
    exit(0)

print("\033[0;36m" + 'Определитель ( $\Delta$ /det): ', get_determinant(a, k),
"\033[0m")
print("\033[0;36m" + 'Определитель ( $\Delta$ /det) из numpy: ', det, "\033[0m")
print_result('Число перестановок: ' + str(k))

print_line()
print("\033[0;34m" + 'Треугольная матрица: ')

for i in range(n):
    print(*a[i], '|', b[i], sep='\t')
print("\033[0m", end="")
print_line()

```

```

print('\033[0;35m' + 'Вектор неизвестных: ')
for i in range(len(x)):
    print("x_", end="")
    print(i, "=", Decimal(str(round(x[i], 5))))
print('\033[0m', end="")
print_line()
print('\033[0;35m' + 'Вектор невязок: ')
for i in range(len(get_r(initial_a, initial_b, x))):
    print("r_", end="")
    print(i, "=", Decimal(str(get_r(initial_a, initial_b, x)[i])))
print('\033[0m', end="")

```

## 5. Примеры работы программы:

```

Вычислительная математика - Лабораторная №1
=====
Каким образом будет осуществлен ввод СЛАУ?
Вручную (+) | С файла 'input.txt' (-): +
Введите размерность матрицы (n): 3
Введите "RANDOM" | "RANDOM" для генерации случайной матрицы (A и B): random
Сгенерированная матрица:
-198 -418 433 | 681
978 -744 109 | 595
226 50 -695 | -910
=====
-542.5819073718433
-2808.6666666666665
-198
Определитель (Δ/det): -301738480.0
Определитель (Δ/det) из numpy: -301738480.0000082
Число перестановок: 0

Треугольная матрица:
-198 -418 433 | 681
0 -2808.6666666666665 2247.757575757576 | 3563.5757575757575
0 0 -542.5819073718433 | -765.919477231369
=====
Вектор неизвестных:
x_0 = 0.34526
x_1 = -0.13907
x_2 = 1.41162
=====
Вектор невязок:
r_0 = 0.0
r_1 = 0.0
r_2 = 1.1368683772161603E-13
Process finished with exit code 0

```

## 6. Вывод:

В рамках этой вводной работы по вычислительной математике я повторно работал с системами линейных алгебраических уравнений и имплементировал их в программной среде. Мне попался метод Гаусса, где откуда я его имплементировал в Питоне. В контексте

вычислительных техников здесь ещё учитываются погрешность решения (в этом случае невязка) и точность решения в случае итерационных методов решения, с которым мне к счастью не пришлось столкнуться.

