

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет ПИ и КТ

Лабораторная работа №1

по дисциплине: «Вычислительная математика»

«Решение системы линейных алгебраических уравнений»

Вариант 1: методом Гаусса

Выполнил:

**Болорболд Аригуун**,

группа P3211

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.

1. **Задание лабораторной работы:**
2. № варианта определяется как номер в списке группы согласно ИСУ. (1)
3. В программе численный метод должен быть реализован в виде отдельной подпрограммы или класса, в который входные/выходные данные передаются в качестве параметров.
4. Размерность матрицы (задаётся из файла или с клавиатуры — по выбору конечного пользователя).
5. 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: ;
* Вывод вектора невязок: ;
* Используя библиотеки выбранного языка программирования найти решение системы линейных уравнений и значение детерминанта. Сравнить результаты;
* Рассмотреть случаи, когда применение метода Гаусса невозможно.

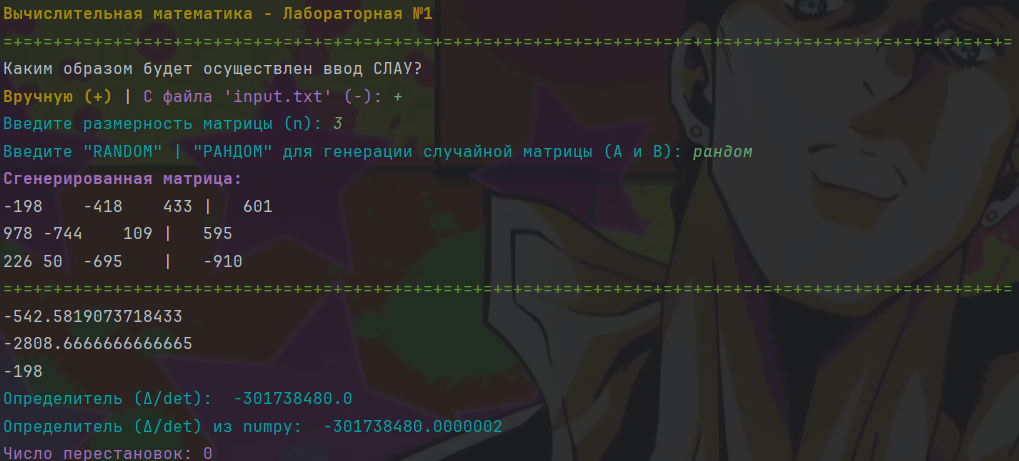
1. **Описание метода, расчётные формулы:**

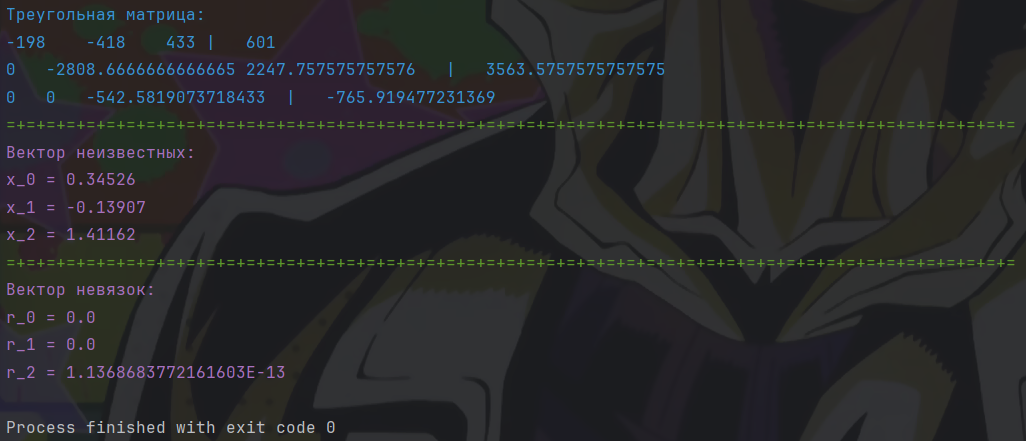
Метод Гаусса — прямой метод, находящий решение путём элементарных преобразования расширенной матрицы системы и приведения в треугольный вид так, чтобы ниже её диагонали находились только нулевые элементы (отсюда и другое название этого метода — последовательного исключения неизвестных), откуда можно получать решение СЛАУ. Здесь требуются не знание формул, а только знание арифметических действии.

1. **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() == 'HFYLJV'.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 строчку ({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 строчку (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 (Δ/det): ', get\_determinant(a, k), "\033[0m")  
print("\033[0;36m" + 'Определитель (Δ/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="")

1. **Примеры работы программы:**





1. **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 оттуда 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.