Университет ИТМО

Лабораторная работа №1 «Системы линейных алгебраических уравнений»

по дисциплине: Вычислительная математика

Вариант: <u>метод Гаусса</u>

Выполнил: Неграш Андрей, P3230

Преподаватель: Перл Ольга Вячеславовна

Санкт-Петербург, 2022

1. Описание метода решения

Метод решения представляет собой преобразование расширенной системы линейных уравнений к треугольному виду и нахождению всех неизвестных, начиная с последнего уравнения.
Когда матрица квадратная и её определитель не равен нулю, то существует лишь одно решение для данной СЛАУ. Каждая следующая неизвестная выражается через предыдущие, начиная с нижней, т.к. она известна сразу.

2. Расчётные формулы

Приведение матрицы в треугольный вид:

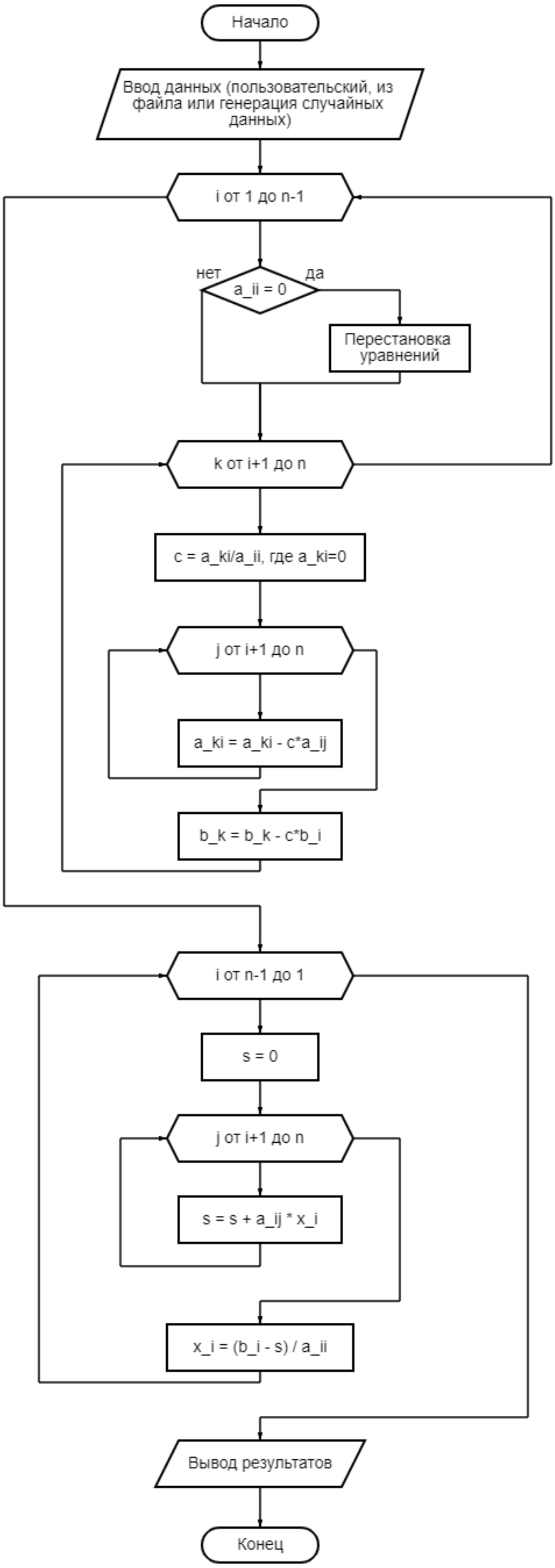
$$\left(\begin{array}{cccc|c} a_{1,1} & a_{1,2} & \cdots & a_{1,n} & b_1 \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} & b_n \end{array}\right) \sim \left(\begin{array}{cccc|c} a_{1,1} & a_{1,2} & \cdots & a_{1,n} & b_1 \\ 0 & a_{2,2} & \cdots & a_{2,n} & b`_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{n,n} & b`_n \end{array}\right)$$

Вычисление неизвестных для обратного хода метода Гаусса:

$x_{n-1} = \frac{b_{n-1}}{a_{n-1,n-1}}$ – для последней неизвестной

$x_i = \frac{b_i - \sum_{j=i+1}^{n-1} a_{i,j} x_j}{a_{i,i}}$ для *i=n-2, n-3, ..., 0*

3. Блок-схема численного метода



4. Листинг численного метода

```
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
            self.swap += 1
            return
        j += 1
    print("Нет решений")
    return ArithmeticError

def __x_calculation(self):
    i = self.n - 2
    self.x.append(self.system[self.n - 1][-1]/self.system[self.n - 1][self.n
- 1])
    while i > -1:
        k = self.n - 1
        value = self.system[i][-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])
        i -= 1

def __make_triangle(self):
    try:
        i = 0
        while i < 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 = 0
            line_sum = 0
            while k < self.n:
                line_sum += self.system[i][k]
                k += 1
            if line_sum == 0:
                print("Данная система уравнений несовместна, решений нет.")
                return ArithmeticError
            i += 1
    except ValueError:
        print("Некорректные данные.")
        return
```

```
def __get_determinate(self):
    i = 0
    self.det = 1
    while i < self.n:
        self.det *= self.system[i][i]
        i += 1
    if self.swap % 2 == 1:
        self.det *= -1
    print("\nОпределитель: " + str(self.det))
    if self.det == 0:
        print("Данная система не имеет решений.")
        return ArithmeticError

def __get_residuals(self):
    i = 0
    print("\nНевязки:")
    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("\tЗначение невязки для уравнения №" + str(i) + " = " +
str(abs(res)))
    print("")
```

Полный код на GitHub: https://github.com/ANegrash/ITMO-all/tree/master/3%20Computational%20math/lab1

5. Работа программы

Лабораторная №1: "Системы линейных алгебраических уравнений"
Вариант: метод Гаусса
Автор: Неграш А.В., P3230

Выберите действие:

1. Пользовательский ввод.
2. Ввод данных из файла.
3. Генерация случайных матриц.
4. Выход.

1

Пожалуйста, введите размер системы уравнений (от 2 до 20):*3*

Вводите строки системы в формате:
 ai1 ai2 ai3 | bi

1. *5 -2 4 | 5*
2. *2 3 -1 | 7*
3. *3 -1 2 | 3*

5.0 x_0 -2.0 x_1 4.0 x_2 | 5.0
2.0 x_0 3.0 x_1 -1.0 x_2 | 7.0
3.0 x_0 -1.0 x_1 2.0 x_2 | 3.0

Треугольная матрица:

5.0 x_0 -2.0 x_1 4.0 x_2 | 5.0
0.0 x_0 3.8 x_1 -2.6 x_2 | 5.0
0.0 x_0 0.0 x_1 -0.26315789473684204 x_2 | -0.26315789473684204

Определитель: -4.999999999999998

Столбец неизвестных:
  X_0: 1.0
  X_1: 2.0
  X_2: 1.0

Невязки:
  Значение невязки для уравнения №1 = 0.0
  Значение невязки для уравнения №2 = 0.0
  Значение невязки для уравнения №3 = 0.0

Выберите действие:
  1. Пользовательский ввод.
  2. Ввод данных из файла.
  3. Генерация случайных матриц.
  4. Выход.

2

В файле должна содержаться матрица следующего формата:

  a11 a12 ... a1n | b1
  a21 a22 ... a2n | b2
  ... ... | ..
  an1 an2 ... ann | bn

Введите путь до файла с данными:*data.txt*

1.0 x_0 1.0 x_1 0.0 x_2 | 0.0
0.0 x_0 1.0 x_1 2.0 x_2 | 4.0
1.0 x_0 -1.0 x_1 1.0 x_2 | 2.0

Треугольная матрица:

1.0 x_0 1.0 x_1 0.0 x_2 | 0.0
0.0 x_0 1.0 x_1 2.0 x_2 | 4.0
0.0 x_0 0.0 x_1 5.0 x_2 | 10.0

Определитель: 5.0

Столбец неизвестных:

  X_0: 0.0
  X_1: 0.0
  X_2: 2.0

Невязки:

  Значение невязки для уравнения №1 = 0.0
  Значение невязки для уравнения №2 = 0.0
  Значение невязки для уравнения №3 = 0.0

Выберите действие:

1. Пользовательский ввод.
2. Ввод данных из файла.
3. Генерация случайных матриц.
4. Выход.

3

Пожалуйста, введите размер системы уравнений (от 2 до 20):*3*

45.67166413617649 x_0 41.32431376360432 x_1 -47.352629617395245 x_2 | -3.5845512296047133
43.51010963155446 x_0 45.43985605324261 x_1 -24.35680652991469 x_2 | 33.285129585698996
42.51383721901681 x_0 34.008705427506854 x_1 -20.796975717033874 x_2 | -20.34870121547334

Треугольная матрица:

45.67166413617649 x_0 41.32431376360432 x_1 -47.352629617395245 x_2 | -3.5845512296047133
0.0 x_0 6.071344826191343 x_1 20.754711628305504 x_2 | 36.70003070502312
0.0 x_0 0.0 x_1 38.522391948122596 x_2 | 9.937910309848206

Определитель: 10681.81326558884

Столбец неизвестных:

X_0: -4.48247696132286
X_1: 5.162905915418975
X_2: 0.25797749846975776

Невязки:

Значение невязки для уравнения №1 = 1.9539925233402755e-14
Значение невязки для уравнения №2 = 0.0
Значение невязки для уравнения №3 = 0.0

Выберите действие:

1. Пользовательский ввод.
2. Ввод данных из файла.
3. Генерация случайных матриц.
4. Выход.

4

Выход. Спасибо за использование этой программы

6. Вывод

В ходе выполнения данной лабораторной работы была создана программа на языке Python для вычисления корней уравнения методом Гаусса. Данный метод является очень хорошим способом решения СЛАУ, так как он более универсален и проще реализуем, однако у простоты есть свои недостатки – мы должны всегда хранить в памяти всю матрицу, что может создать проблемы при решении матриц с большими размерами. Также существует проблема с погрешностями, поскольку мы работаем с результатами предыдущих математических операций, которые были подвергнуты округлениям.

В отличие от метода Гаусса, метод Гаусса с выбором главного элемента имеет меньшую погрешность, так как она накапливается только во время приведения матрицы к треугольному виду, в остальном же и алгоритмическая сложность одинаковая, как и проблемы с объёмом памяти.

Если сравнивать прямые и итерационные методы, то для итерационных методов есть возможность задать точность вычислений и таким образом значительно снизить её погрешность, также нет необходимости хранить всю матрицу в памяти, однако есть проблема – для больших матриц сложнее найти такую, которая бы удовлетворяла условию сходимости, так что универсальными эти методы не назовёшь. Также в случаях для большого количества итераций алгоритмическая сложность итерационных методов возрастает и может начать превышать сложность для прямых методов.