Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

09.03.04 Программная инженерия

Системное и прикладное программное обеспечение

Лабораторная работа №1
По дисциплине «Вычислительная математика»
Вариант № 1

Выполнила студентка группы P3213:

Авшистер Ольга Аркадьевна

Преподаватель:

Машина Екатерина Алексеевна

г. Санкт-Петербург 2024 г.

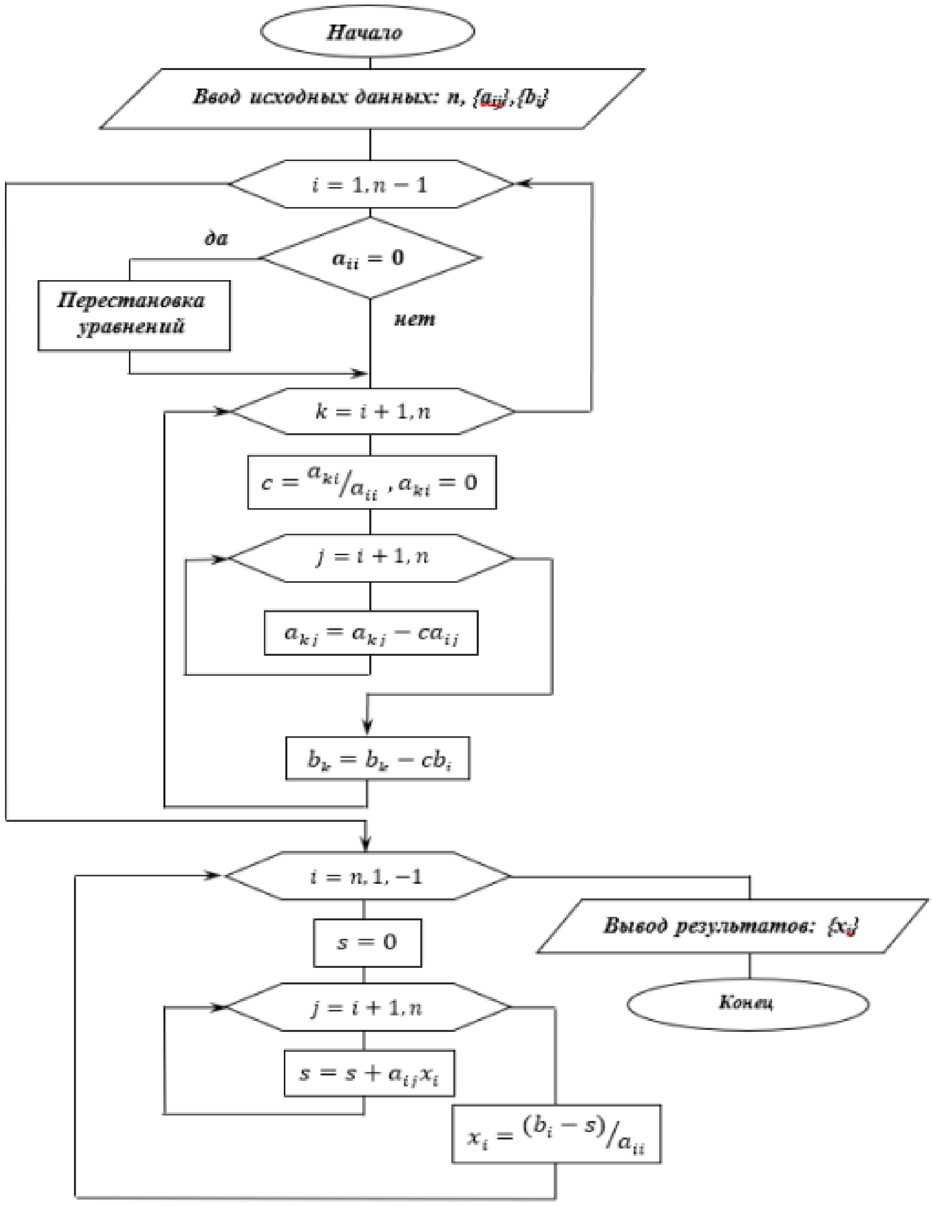
Цель работы

Изучить различные методы решения системы линейных алгебраических уравнений, их особенности, достоинства и недостатки. Реализовать метод Гаусса решения СЛАУ и протестировать его на различных входных данных.

Описание метода, расчетные формулы

Метод состоит из двух частей: прямого и обратного ходов. Прямой ход метода Гаусса состоит в последовательном исключении неизвестных из уравнений системы. Обратный ход метода Гаусса состоит в последовательном вычислении искомых неизвестных. Использовать данный метод можно только в случае, когда определитель матрицы не равен нулю.

Пример блок-схемы метода Гаусса:



Листинг программы

```
def gauss(m, d):
    n = len(m)
    if d == 0:
        print("Метод Гаусса нельзя использовать для матриц с нулевым
определителем")
        return 0
    # ПРЯМОЙ
    for k in range(n):
        if m[k][k] == 0:
            dop = m[k]
            for i in range(k+1, n):
                if m[i][k] != 0:
                    m[k] = m[i]
                    m[i] = dop
                    break
        for j in range(k + 1, n):
            for i in range(n, k - 1, -1):
                m[j][i] -= m[k][i] * m[j][k] / m[k][k]
    print("Треугольная матрица:")
    for i in m:
        print([float('%.2f' % elem) for elem in i])
    # ОБРАТНЫЙ
    v = [0 for i in range(n)]
    for i in range(n):
        v[n - i - 1] = (m[n - i - 1][n] - sum(m[n - i - 1][n - i + j] * v[n -
i + j] for j in range(i))) / m[n - i - 1][n - i - 1]
    r = [0 for i in range(n)]
    for i in range(n):
        for j in range(n):
            r[i] += m[i][j] * v[j]
        r[i] -= m[i][n]
    print("Вектор неизвестных: ", [float('%.2f' % elem) for elem in v])
    print("Вектор невязок: ", [float('%.2f' % elem) for elem in r])
    return 1
```

Примеры и результаты работы программы

```
Введите 1, если ввод данных будет происходить из файла. Введите 2, если с клавиатуры. Введите 3 для генерации рандомной матрицы:
Размерность:
4
Матрица:
[1.26, 5.96, 5.26, 1.89, 4.77]
[0.11, 8.26, 6.65, 8.31, 0.82]
[3.61, 7.46, 6.23, 5.77, 5.03]
[1.91, 8.35, 7.97, 0.36, 7.68]
Определитель равен  -35.55
Треугольная матрица:
[1.26, 5.96, 5.26, 1.89, 4.77]
[0.0, 7.74, 6.19, 8.15, 0.4]
[0.0, 0.0, -1.15, 10.47, -8.14]
[-0.0, 0.0, 0.0, 3.18, -3.37]
Вектор неизвестных:  [0.86, 3.24, -2.59, -1.06]
Вектор невязок:  [-0.0, -0.0, 0.0, 0.0]
```

```
Введите 1, если ввод данных будет происходить из файла. Введите 2, если с клавиатуры. Введите 3 для генерации рандомной матрицы: 2
Введите размерность матрицы (<=20)
3
Введите матрицу
4 9 8 5
1 6 7 5
3 2 4 8
Определитель равен  65.0
Треугольная матрица:
[4.0, 9.0, 8.0, 5.0]
[0.0, 3.75, 5.0, 3.75]
[0.0, 0.0, 4.33, 9.0]
Вектор неизвестных:  [1.08, -1.77, 2.08]
Вектор невязок:  [0.0, 0.0, 0.0]
```

Вывод

Как и у любого другого метода решения СЛАУ, у метода Гаусса есть свои достоинства и недостатки. Данный метод является точным и менее трудоемким по сравнению с другими методами, но только если мы говорим про матрицы с небольшой размерностью. С увеличением порядка системы общее количество действий растет очень стремительно. А также нужно постоянно следить за элементами главной диагонали и менять строки местами, если встречается нулевой элемент. Использовать данный метод можно только в случае, когда определитель матрицы не равен нулю.

Если сравнивать данный метод с итерационными методами, можно сказать, что их алгоритмы более сложные, однако в них не накапливаются погрешности, тк точность вычислений в каждой итерации определяется лишь результатами предыдущей итерации и практически не зависит от ранее выполненных вычислений.