

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АДЫГЕЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Инженерно-физический факультет
Кафедра автоматизированных систем обработки информации и
управления

ОТЧЕТ ПО ПРАКТИКЕ

Программная реализация численного метода
*Решение системы линейных алгебраических
уравнений методом Гаусса-Жордана.*

1 курс, группа 1ИВТ1 (1)

Выполнил:

_____ К. Д. Попов
«___» _____ 2023 г.

Руководитель:

_____ С. В. Теплоухов
«___» _____ 2023 г.

Майкоп, 2023 г.

1. Введение

1.1. Формулировка цели

Целью данной работы является написание программы для решения системы линейных алгебраических уравнений методом Гаусса-Жордана.

1.1.1. Теория

Нахождение ранга матрицы способом элементарных преобразований (методом Гаусса). Под элементарными преобразованиями матрицы понимаются следующие операции

1.1.2. Алгоритм

- 1) Выбирают первый слева столбец матрицы, в котором есть хоть одно отличное от нуля значение.
- 2) Если самое верхнее число в этом столбце ноль, то меняют всю первую строку матрицы с другой строкой матрицы, где в этой колонке нет нуля.
- 3) Все элементы первой строки делят на верхний элемент выбранного столбца.
- 4) Из оставшихся строк вычитают первую строку, умноженную на первый элемент соответствующей строки, с целью получить первым элементом каждой строки (кроме первой) ноль.
- 5) Далее проводят такую же процедуру с матрицей, получающейся из исходной матрицы после вычёркивания первой строки и первого столбца.
- 6) После повторения этой процедуры $n-1$ раз получают верхнюю треугольную матрицу.
- 7) Вычитают из предпоследней строки последнюю строку, умноженную на соответствующий коэффициент, с тем, чтобы в предпоследней строке осталась только 1 на главной диагонали.
- 8) Повторяют предыдущий шаг для последующих строк. В итоге получают единичную матрицу и решение на месте свободного вектора (с ним необходимо проводить все те же преобразования).

2. Ход работы

2.1. Код выполненной программы

```
import numpy as np
# исходная матрица
A = np.array([[2, 1, -1, 8],
              [-3, -1, 2, -11],
              [-2, 1, 2, -3]])
# свободный вектор
b = np.array([3, -6, -1])
# объединяем матрицу и свободный вектор в одну матрицу
Ab = np.column_stack((A, b))
# получаем размеры матрицы Ab
n, m = Ab.shape
# проходимся по столбцам матрицы Ab
for j in range(n-1):
    # выбираем первый ненулевой элемент в столбце
    for i in range(j, n):
        if Ab[i,j] != 0:
            break
    else:
        # если все элементы в столбце нули, переходим к следующему столбцу
        continue

    # меняем местами строки i и j
    Ab[[j,i], :] = Ab[[i,j], :]

    # делим первую строку на Ab[j,j]
    Ab[j, :] /= Ab[j,j]

    # вычитаем первую строку из остальных строк
    for i in range(n):
        if i == j:
            continue
        Ab[i, :] -= Ab[i,j] * Ab[j, :]

# выводим результат
x = Ab[:, -1]
print('Решение: x =', x)
```

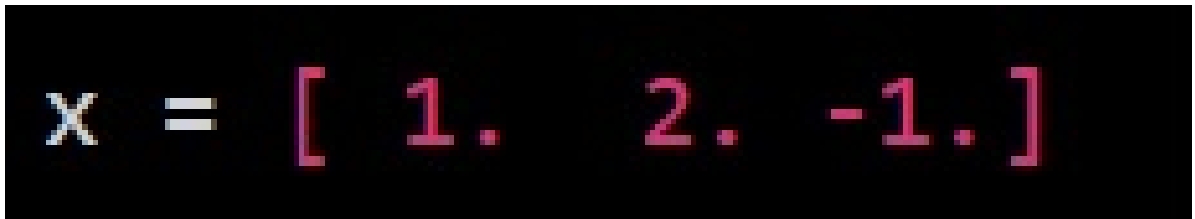

$$x = [1. \quad 2. \quad -1.]$$

Рис. 1. Результат работы