

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 обратной матрицы.
Вариант 5.

1 курс, группа ИИВТ2-2

Выполнила:

_____ Е. Е. Бубенщикова
«___» _____ 2023 г.

Руководитель:

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

Майкоп, 2023 г.

1. Введение

1.1. Формулировка цели

Целью данной работы является написание программы для вычисления матрицы обратную заданной.

1.1.1. Теория

Нахождение обратной матрицы методом исключения неизвестных Гаусса. Первый шаг для нахождения обратной матрицы методом исключения неизвестных Гаусса - приписать к матрице A единичную матрицу того же порядка, отделив их вертикальной чертой. Мы получим сдвоенную матрицу $A|E$. Умножим обе части этой матрицы на A^{-1} , тогда получим $(A \cdot A^{-1} | E \cdot A^{-1})$, но $A \cdot A^{-1} = E$ и $E \cdot A^{-1} = A^{-1}$. Алгоритм нахождения обратной матрицы методом исключения неизвестных Гаусса

- 1) К матрице A приписать единичную матрицу того же порядка;
- 2) Полученную сдвоенную матрицу преобразовать так, чтобы в левой её части получилась единичная матрица, тогда в правой части на месте единичной матрицы автоматически получится обратная матрица. Матрица A в левой части преобразуется в единичную матрицу путём элементарных преобразований матрицы;
- 3) Если в процессе преобразования матрицы A в единичную матрицу в какой-либо строке или в каком-либо столбце окажутся только нули, то определитель матрицы равен нулю, и, следовательно, матрица A будет вырожденной, и она не имеет обратной матрицы. В этом случае дальнейшее нахождение обратной матрицы прекращается.

2. Ход работы

2.1. Код выполненной программы

```
import numpy as np

# Считываем размерность матрицы
n = int(input("Введите количество строк: "))
m = int(input("Введите количество столбцов: "))

# Считываем элементы матрицы
A = np.zeros((n, m))
for i in range(n):
```

```

    for j in range(m):
        A[i, j] = float(input( "Введите элемент [{i + 1}, {j + 1}]))

# Вычисляем обратную матрицу
det = np.linalg.det(A)
if det == 0:
    print("Матрица необратима")
else:
    A_inv = np.linalg.inv(A)
    if m == n:
        # Если матрица квадратная, выводим обратную матрицу
        print("Обратная матрица: ")
        print(A_inv)
    else:
        # Если матрица прямоугольная, выводим обратную матрицу с округлением до 2 зна
        A_inv = np.round(A_inv, 2)
        print("Обратная матрица: ")
        print(A_inv)

```

```

Введите количество строк: 4
Введите количество столбцов: 4
Введите элемент  $[[i + 1], [j + 1]]$  3
Введите элемент  $[[i + 1], [j + 1]]$  7
Введите элемент  $[[i + 1], [j + 1]]$  5
Введите элемент  $[[i + 1], [j + 1]]$  2
Введите элемент  $[[i + 1], [j + 1]]$  3
Введите элемент  $[[i + 1], [j + 1]]$  5
Введите элемент  $[[i + 1], [j + 1]]$  7
Введите элемент  $[[i + 1], [j + 1]]$  4
Введите элемент  $[[i + 1], [j + 1]]$  6
Введите элемент  $[[i + 1], [j + 1]]$  3
Введите элемент  $[[i + 1], [j + 1]]$  2
Введите элемент  $[[i + 1], [j + 1]]$  5
Введите элемент  $[[i + 1], [j + 1]]$  7
Введите элемент  $[[i + 1], [j + 1]]$  6
Введите элемент  $[[i + 1], [j + 1]]$  3
Введите элемент  $[[i + 1], [j + 1]]$  4
Обратная матрица:
[[-0.69293246  0.30701754  0.60421053  0.09473604]
 [ 0.63157025  0.36042105  0.42105263  0.47360421]
 [-0.53508772  0.46491228 -0.57894737  0.52631579]
 [ 0.66666667 -0.33333333  1.          -1.          ]]

```

Рис. 1. Результат работы