

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
Нахождение обратной матрицы методом
исключения неизвестных Гаусса.

2 курс, группа 2ИВТ

Выполнил:

_____ Ю. А. Береснев
«___» _____ 2021 г.

Руководитель:

_____ С. В. Теплоухов
«___» _____ 2021 г.

Майкоп, 2021 г.

1. Введение

1.1. Цель работы

Целью данной работы является вычисление матрицы обратной заданной.

1.2. 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 $E \cdot A^{-1} = A^{-1}$.

2. Ход работы

2.1. Код программы

```
#include <conio.h>
#include <iostream>
#include <math.h>
using namespace std;
int opr(double** a, int n) {
    if (n == 1)
        return a[0][0];
    else if (n == 2)
        return a[0][0] * a[1][1] - a[0][1] * a[1][0];
    else {
        double d = 0;
        for (int k = 0; k < n; k++) {
            double** m = new double* [n - 1];
            for (int i = 0; i < n - 1; i++) {
                m[i] = new double[n - 1];
            }
            for (int i = 1; i < n; i++) {
                int t = 0;
                for (int j = 0; j < n; j++) {
                    if (j == k)
                        continue;
                    m[i - 1][t] = a[i][j];
                    t++;
                }
            }
            d += pow(-1, k + 2) * a[0][k] * opr(m, n - 1);
        }
    }
}
```

```

        }
        return d;
    }
}

void obr(double** O, int n)
{
    double temp;
    double** E = new double* [n];

    for (int i = 0; i < n; i++)
        E[i] = new double[n];

    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
        {
            if (i == j) E[i][j] = 1.0;
            else E[i][j] = 0.0;
        }

    for (int k = 0; k < n; k++)
    {
        temp = O[k][k];

        for (int j = 0; j < n; j++)
        {
            O[k][j] /= temp;
            E[k][j] /= temp;
        }

        for (int i = k + 1; i < n; i++)
        {
            temp = O[i][k];

            for (int j = 0; j < n; j++)
            {
                O[i][j] -= O[k][j] * temp;
                E[i][j] -= E[k][j] * temp;
            }
        }
    }

    for (int k = n - 1; k > 0; k--)
    {
        for (int i = k - 1; i >= 0; i--)

```

```

        {
            temp = O[i][k];

            for (int j = 0; j < n; j++)
            {
                O[i][j] -= O[k][j] * temp;
                E[i][j] -= E[k][j] * temp;
            }
        }
    }
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            O[i][j] = E[i][j];
}

int main()
{
    setlocale(LC_ALL, "Russian");
    int n;
    cout << "Укажите размерность квадратной матрицы:" << endl << "n=";
    cin >> n;
    double** A = new double* [n];
    cout << "Введите элементы матрицы:" << endl;
    for (int i = 0; i < n; i++) {
        A[i] = new double[n];
        for (int j = 0; j < n; j++) {
            cout << "A[" << i << "][" << j << "]=";
            cin >> A[i][j];
        }
    }
    if (n < 1) cout << "Определитель вычислить невозможно и обратной матрицы не существует";
    else
    {
        cout << "Определитель для заданной матрицы = " << opr(A, n) << endl;
        if (opr(A, n) == 0) cout << "Матрица вырожденная и обратной матрицы для неё не существует";
        else {
            obr(A, n);
            cout << "Обратная матрица для заданной:" << endl;
            for (int i = 0; i < n; i++) {
                for (int j = 0; j < n; j++) {
                    cout << A[i][j] << " ";
                } cout << endl;
            }
        }
    }
}

```

```

_getch();
return 0;
}

```

```

C:\Users\beres\source\repos\Project1\Debug\Project1.exe
Укажите размерность квадратной матрицы:
n=3
Введите элементы матрицы:
A[0][0]=1
A[0][1]=2
A[0][2]=3
A[1][0]=3
A[1][1]=2
A[1][2]=3
A[2][0]=1
A[2][1]=2
A[2][2]=3
Определитель для заданной матрицы = 0
Матрица вырожденная и обратной матрицы для неё не существует

```

Рис. 1. Окно программы с вырожденной матрицей

```

C:\Users\beres\source\repos\Project1\Debug\Project1.exe
Укажите размерность квадратной матрицы:
n=3
Введите элементы матрицы:
A[0][0]=3
A[0][1]=2
A[0][2]=3
A[1][0]=2
A[1][1]=1
A[1][2]=3
A[2][0]=3
A[2][1]=1
A[2][2]=2
Определитель для заданной матрицы = 4
Обратная матрица для заданной:
-0.25 -0.25 0.75
1.25 -0.75 -0.75
-0.25 0.75 -0.25

```

Рис. 2. Окно программы с обратной матрицей

Список литературы

- [1] Кнут Д.Э. Всё про $\text{T}_\text{E}\text{X}$. — Москва: Изд. Вильямс, 2003 г. 550 с.
- [2] Львовский С.М. Набор и верстка в системе $\text{L}_\text{A}\text{T}_\text{E}\text{X}$. — 3-е издание, исправленное и дополненное, 2003 г.
- [3] Воронцов К.В. $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ в примерах. 2005 г.