

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
НИЖЕГОРОДСКИЙ ГОССУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. Р. Е. АЛЕКСЕЕВА

Кафедра «Прикладная математика»

Отчет
Лабораторная работа №1
по дисциплине «Численные Методы»

Тема: «СЛАУ»

Студент

(Подпись) Шохов М.Е.
(Фамилия, И., О.)

17-ПМ 31.10.19
(Группа) (Дата сдачи)

Проверила

(Подпись) Талалушкина Л.В.
(Фамилия, И., О.)

Отчет защищен «__» _____ 2019г.

с оценкой _____

Нижний Новгород, 2019

Оглавление

1.	ЦЕЛЬ РАБОТЫ.....	3
2.	ПОЯСНЕНИЕ К ЗАДАНИЮ.....	4
3.	ОПИСАНИЕ АЛГОРИТМА.....	6
4.	РЕАЛИЗАЦИЯ НА ЯЗЫКЕ C++.....	7
5.	РЕАЛИЗАЦИЯ НА ЯЗЫКЕ PYTHON.....	13
6.	РЕАЛИЗАЦИЯ НА ЯЗЫКЕ FORTRAN.....	16
7.	РЕАЛИЗАЦИЯ НА ЯЗЫКЕ MATLAB	21
8.	РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММ	25
9.	ВЫВОД.....	28

1	Вып.	Шохов М.Е.		6.11.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		6.11.19		2
№.		Ф.И.О.	Подп.	Дата		

1. ЦЕЛЬ РАБОТЫ

Изучить и программно реализовать метод Гаусса – метод решения системы линейных алгебраических уравнений, на языках высокого уровня (C, Python, Matlab, Fortran).

1	Вып.	Шохов М.Е.		6.11.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		6.11.19		3
№.		Ф.И.О.	Подп.	Дата		

2. ПОЯСНЕНИЕ К ЗАДАНИЮ

Метод Гаусса был предложен известнейшим немецким математиком Карлом Фридрихом Гауссом (1777 - 1855) и является одним из наиболее универсальных методов решения СЛАУ. Сущность этого метода состоит в том, что посредством последовательных исключений неизвестных данная система превращается в ступенчатую (в частности, треугольную) систему, равносильную данной. При практическом решении задачи, расширенная матрица системы с помощью элементарных преобразований над ее строками приводится к ступенчатому виду. Далее последовательно находятся все неизвестные, начиная снизу вверх.

Метод Гаусса включает в себя прямой (приведение расширенной матрицы к ступенчатому виду, то есть получение нулей под главной диагональю) и обратный (получение нулей над главной диагональю расширенной матрицы) ходы. Прямой ход и называется методом Гаусса, обратный - методом Гаусса-Жордана, который отличается от первого только последовательностью исключения переменных.

Метод Гаусса идеально подходит для решения систем содержащих больше трех линейных уравнений, для решения систем уравнений, которые не являются квадратными (чего не скажешь про метод Крамера и матричный метод). То есть метод Гаусса - наиболее универсальный метод для нахождения решения любой системы линейных уравнений, он работает в случае, когда система имеет бесконечно много решений или несовместна.

Последовательность действий для общего решения системы уравнения методом Гаусса заключается в поочередном применении прямого и обратного хода к матрице на основе СЛАУ. Пусть исходная система уравнений имеет следующий вид:

$$\begin{cases} a_{11} \cdot x_1 + \dots + a_{1n} \cdot x_n = b_1 \\ \dots \\ a_{m1} \cdot x_1 + a_{mn} \cdot x_n = b_m \end{cases}$$

Чтобы решить СЛАУ методом Гаусса, необходимо записать исходную систему уравнений в виде матрицы:

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \dots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}, b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$$

Матрица A называется основной матрицей и представляет собой записанные по порядку коэффициенты при переменных, а b называется столбцом её свободных членов.

1	Вып.	Шохов М.Е.		6.11.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		6.11.19		4
№.		Ф.И.О.	Подп.	Дата		

Матрица A , записанная через черту со столбцом свободных членов называется расширенной матрицей:

$$A = \begin{array}{ccc|c} a_{11} & \dots & a_{1n} & b_1 \\ \vdots & \dots & \vdots & \dots \\ a_{m1} & \dots & a_{mn} & b_m \end{array}$$

Теперь необходимо с помощью элементарных преобразований над системой уравнений (или над матрицей, так как это удобнее) привести её к следующему виду:

$$\left\{ \begin{array}{l} \alpha_{1j_1} \cdot x_{j_1} + \alpha_{1j_2} \cdot x_{j_2} \dots + \alpha_{1j_r} \cdot x_{j_r} + \dots \alpha_{1j_n} \cdot x_{j_n} = \beta_1 \\ \alpha_{2j_2} \cdot x_{j_2} \dots + \alpha_{2j_r} \cdot x_{j_r} + \dots \alpha_{2j_n} \cdot x_{j_n} = \beta_2 \\ \dots \\ \alpha_{rj_r} \cdot x_{j_r} + \dots \alpha_{rj_n} \cdot x_{j_n} = \beta_r \\ 0 = \beta_{(r+1)} \\ \dots \\ 0 = \beta_m \end{array} \right. \quad (1)$$

Матрица, полученная из коэффициентов преобразованной системы уравнения (1) называется ступенчатой.

1	Вып.	Шохов М.Е.		6.11.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		6.11.19		5
№.		Ф.И.О.	Подп.	Дата		

3. ОПИСАНИЕ АЛГОРИТМА

Основные функции:

- initializationFromUserInput() – ввод матрицы пользователем через консоль,
- initializationFromFile() – чтение матрицы из файла «matrix.txt»,
- showSystem() – отображение матрицы в консоли,
- applyGaussMethod() – осуществление элементарных преобразований над матрицей (перемена строк местами, получение треугольной матрицы) и вычисление решения СЛАУ,
- showSolutions() – отображение решения СЛАУ в консоли.

В функции main() объявляется объект класса (в случае Fortran - модуля) GaussMethod, а затем идёт поочерёдный вызов методов: считывания (один из двух возможных способов), отображения введённой системы, применения метода Гаусса, вывода на экран решения СЛАУ.

Основные элементы:

- systemSize - количество уравнений системы
- coeff - массив коэффициентов при неизвестных
- freeTerm - массив свободных членов
- solutions - массив решений заданной системы
- filename - имя файла, в котором хранится СЛАУ («matrix.txt»)
- isSolutionFounded - флаг, обозначающий: найдено решение или нет
- rankOfUsualMatrix - ранг обычной матрицы
- rankOfExtendedMatrix - ранг расширенной матрицы

Вывод программы: введённая (считанная из файла) система, сообщение о наличии решения: нет решения, система имеет бесконечное множество решений, система имеет единственное решение. Если система имеет решение, то помимо сообщения об этом, будет выведено оно само.

Исходная матрица					
№	Матрица коэффициентов A				Вектор b
15	3.81	0.25	1.28	0.75	4.21
	2.25	1.32	4.58	0.49	6.47
	5.31	6.28	0.98	1.04	2.38
	9.39	2.45	3.35	2.28	10.48

1	Вып.	Шохов М.Е.		6.11.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		6.11.19		6
№.		Ф.И.О.	Подп.	Дата		

4. РЕАЛИЗАЦИЯ НА ЯЗЫКЕ C++

Реализация на языке C++

```
#include <iostream>
#include <fstream>
#include <string>
#include <math.h>
#include <utility>
#include <Eigen/Dense>

using namespace Eigen;
using namespace std;

class GaussMethod{
private:
    int systemSize;
    double** coeff;
    double** tmpMatrix;
    double* freeTerm;
    double* solutions;
    string filename;
    bool isSolutionFounded;
public:
    GaussMethod();
    ~GaussMethod();
    long int getRank(double**, int, int);
    void swap(double**, int, int, int);
    void initializeFromFile();
    void initializeFromUserInput();
    void showSystem();
    void applyGausMethod();
    void showSolutions();
    void deleteTmp();
    double** getCopy(bool);
};

GaussMethod::GaussMethod()
{
    filename = "matrix.txt";
    isSolutionFounded = false;
}

GaussMethod::~~GaussMethod()
{
    delete [] solutions;
    for (int i = 0; i < systemSize; i++)
        delete [] coeff[i];
    delete [] coeff;
    delete [] freeTerm;
}
```

1	Вып.	Шохов М.Е.		6.11.19
2	Пров.	Талалушкина Л.В.		6.11.19
№.		Ф.И.О.	Подп.	Дата

ЛР по «Численным методам»-НГТУ-(17-ПМ)

Лист

7

Реализация на языке C++

```

long int GaussMethod::getRank(double** a, int nSize, int mSize)
{
    MatrixXd matrix(nSize, mSize);
    for(int i = 0; i < nSize; i++)
    {
        for(int j = 0; j < mSize; j++)
        {
            matrix(i,j) = a[i][j];
        }
    }

    FullPivLU<MatrixXd> lu_decomp(matrix);
    long int rank = lu_decomp.rank();

    return rank;
}

void GaussMethod::swap(double** mat, int row1, int row2, int col)
{
    for (int i = 0; i < col; i++)
    {
        double temp = mat[row1][i];
        mat[row1][i] = mat[row2][i];
        mat[row2][i] = temp;
    }
}

void GaussMethod::initializeFromFile()
{
    ifstream fout(filename);

    fout >> systemSize;

    coeff = new double*[systemSize];
    freeTerm = new double[systemSize];

    for (int i = 0; i < systemSize; i++)
    {
        coeff[i] = new double[systemSize+1];
        for (int j = 0; j < systemSize; j++)
        {
            fout >> coeff[i][j];
        }

        fout >> freeTerm[i];
    }
}
    
```


Реализация на языке C++

```

fout.close();
}

void GaussMethod::initializeFromUserInput()
{
    cout << "Введите количество уравнений: ";
    cin >> systemSize;
    coeff = new double*[systemSize];
    freeTerm = new double[systemSize];

    cout << "Введите коэффициенты: \n";
    for (int i = 0; i < systemSize; i++)
    {
        cout << i+1 << "уравнение: \n";
        coeff[i] = new double[systemSize];
        for (int j = 0; j < systemSize; j++)
        {
            cout << "a[" << i << "][" << j << "] = ";
            cin >> coeff[i][j];
        }

        cout << "Введите свободный член: \n";
        cout << "y[" << i << "] = ";
        cin >> freeTerm[i];
    }
}

void GaussMethod::showSystem()
{
    cout << "Введённая система: \n";
    for (int i = 0; i < systemSize; i++)
    {
        for (int j = 0; j < systemSize; j++)
        {
            if (coeff[i][j] < 0)
                cout << "(" << coeff[i][j] << ") * x" << j;
            else
                cout << coeff[i][j] << " * x" << j;
            if (j < systemSize - 1)
                cout << " + ";
        }
        cout << " = " << freeTerm[i] << endl;
    }
    return;
}

```

Реализация на языке C++

```
void GaussMethod::applyGausMethod()
{
    solutions = new double[systemSize];

    long int rankOfUsualMatrix = getRank(getCopy(0),systemSize, systemSize);
    deleteTmp();

    double x[systemSize];

    for (int i=0; i<systemSize; i++)
        coeff[i][systemSize] = freeTerm[i];

    long int rankOfExtendedMatrix = getRank(getCopy(1),systemSize,systemSize+1);
    deleteTmp();

    if (rankOfUsualMatrix < rankOfExtendedMatrix)
    {
        cout << "Нет решения!" << endl;
        return;
    }
    else if (rankOfUsualMatrix == rankOfExtendedMatrix && rankOfUsualMatrix <
systemSize)
    {
        cout << "Система имеет бесконечное множество решений!" << endl;
        return;
    }
    else if (rankOfUsualMatrix == rankOfExtendedMatrix && rankOfUsualMatrix ==
systemSize)
        cout << "Система имеет единственное решение!" << endl;

    for(int i=systemSize-1; i>0; i--)
    {
        if(coeff[i-1][0]<coeff[i][0])
        {
            for(int j=0; j<=systemSize; j++)
            {
                double c = coeff[i][j];
                coeff[i][j] = coeff[i-1][j];
                coeff[i-1][j] = c;
            }
        }
    }
}
```

Реализация на языке C++

```

for(int k=0; k<systemSize-1; k++)
    for(int i=k; i<systemSize-1; i++)
    {
        double c = (coeff[i+1][k]/coeff[k][k]) ;

        for(int j=0; j<=systemSize; j++)
            coeff[i+1][j]-=c*coeff[k][j];
    }

for(int i=systemSize-1; i>=0; i--)
{
    double c = 0;
    for(int j=i; j<=systemSize-1; j++)
        c = c+coeff[i][j]*x[j];

    x[i]=(coeff[i][systemSize]-c)/coeff[i][i];
}

isSolutionFounded = true;

for (int i=0; i<systemSize; i++)
    solutions[i] = x[i];
}

void GaussMethod::showSolutions()
{
    if (isSolutionFounded)
    {
        cout << "Решение введенной системы: \n";
        for (int i = 0; i < systemSize; i++)
            cout << "x[" << i << "]=" << solutions[i] << endl;
    }
}

void GaussMethod::deleteTmp()
{
    for (int i = 0; i < systemSize; i++)
        delete [] tmpMatrix[i];
    delete [] tmpMatrix;
}

```

1	Вып.	Шохов М.Е.		6.11.19
2	Пров.	Талалушкина Л.В.		6.11.19
№.		Ф.И.О.	Подп.	Дата

Реализация на языке C++

```
double **GaussMethod::getCopy(bool mode)
{
    tmpMatrix = new double*[systemSize];
    int nSize = 0;
    int mSize = systemSize;
    mode ? nSize = systemSize + 1 : nSize = systemSize;

    for (int i = 0; i < mSize; i++)
    {
        tmpMatrix[i] = new double[nSize];
        for (int j = 0; j < nSize; j++)
            tmpMatrix[i][j] = coeff[i][j];
    }

    return tmpMatrix;
}

int main(int arg, char** argv)
{
    GaussMethod gaussMethod;

    if(strcmp(argv[1], "user") == 0)
        gaussMethod.initializeFromUserInput();
    else if (strcmp(argv[1], "file") == 0)
        gaussMethod.initializeFromFile();

    gaussMethod.showSystem();
    gaussMethod.applyGausMethod();
    gaussMethod.showSolutions();

    return 0;
}
```

5. РЕАЛИЗАЦИЯ НА ЯЗЫКЕ PYTHON

Реализация на языке Python

```
import numpy as np
from sys import argv

class GaussMethod :

    def __init__ (self) :
        self.systemSize = 0
        self.coeff = []
        self.freeTerm = []
        self.solutions = []
        self.filename = "matrix.txt"
        self.isSolutionFounded = False

    def initializationFromUserInput(self):
        self.systemSize = int(input("Введите количество уравнений:
"))

        print("Введите коэффициенты: ")
        for i in range(self.systemSize) :
            print(f"{i+1} уравнение: ")
            self.coeff.append([])

            for j in range(self.systemSize) :
                self.coeff[i].append(float(input(f"a[{i}][{j}]= ")))

            print("Введите свободный член: ")
            self.freeTerm.append(float(input(f"y[{i}]= ")))

        self.solutions = [None for i in range(0,self.systemSize)]

    def initializationFromFile(self) :
        with open('matrix.txt') as file :
            self.coeff = [list(map(float, row.split())) for row in
file.readlines()]
            self.systemSize = len(self.coeff[0])-1

            for i in range(0,self.systemSize) :
                self.freeTerm.append(self.coeff[i][-1])
                self.coeff[i].pop()

            self.solutions = [None for i in range(0,self.systemSize)]

    def showSystem(self):
        print("Введённая система: ")
        for row in range(len(self.freeTerm)):
            for col in range(len(self.coeff[row])):
                print(f"{self.coeff[row][col]}*x{col}", end='')

```

1	Вып.	Шохов М.Е.		6.11.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		6.11.19		13
№.		Ф.И.О.	Подп.	Дата		

Реализация на языке Python

```

        if col < self.systemSize - 1 :
            print(" + ", end='')
        print(f" = {self.freeTerm[row]}")

def applyGaussMethod (self):
    x = [0 for i in range(0,self.systemSize)]

    tmpNpArr = np.array(self.coeff)
    rankOfUsualMatrix = np.linalg.matrix_rank(tmpNpArr)
    # determinant = np.linalg.det(self.coeff)

    for i in range(0,self.systemSize) :
        self.coeff[i].append(self.freeTerm[i])

    tmpNpArr = np.array(self.coeff)
    rankOfExtendedMatrix = np.linalg.matrix_rank(tmpNpArr)

    if rankOfUsualMatrix < rankOfExtendedMatrix :
        print("Нет решения!")
        return
    elif rankOfUsualMatrix == rankOfExtendedMatrix and
rankOfUsualMatrix < self.systemSize :
        print("Система имеет бесконечное множество решений! ")
        return
    elif rankOfUsualMatrix == rankOfExtendedMatrix and
rankOfUsualMatrix == self.systemSize :
        print("Система имеет единственное решение!")

    for i in range(self.systemSize-1,0,-1) :
        if self.coeff[i-1][0] < self.coeff[i][0] :
            for j in range(0,self.systemSize+1) :
                self.coeff[i][j], self.coeff[i-1][j] =
self.coeff[i-1][j], self.coeff[i][j]

    for k in range(0,self.systemSize-1) :
        for i in range(k,self.systemSize-1) :
            tmp = (self.coeff[i+1][k] / self.coeff[k][k])

            for j in range(0,self.systemSize+1) :
                self.coeff[i+1][j] -= tmp * self.coeff[k][j]

    for i in range(self.systemSize-1,-1,-1) :
        tmp = 0.0
        for j in range(i,self.systemSize) :
            tmp = tmp + self.coeff[i][j]*x[j]
        x[i] = (self.coeff[i][self.systemSize]-tmp) /
self.coeff[i][i]

    self.isSolutionFounded = True

```

1	Вып.	Шохов М.Е.		6.11.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		6.11.19		14
№.		Ф.И.О.	Подп.	Дата		

Реализация на языке Python

```

        for i in range(0,self.systemSize) :
            self.solutions[i] = x[i];
        # self.solutions = x

    def showSolutions(self) :
        if self.isSolutionFounded :
            print("Решение введённой системы: ");
            for i in range(self.systemSize) :
                print(f"x[{i}]= {self.solutions[i]}")

if __name__ == "__main__" :
    scriptName, mode = argv

    gaussMethod = GaussMethod()
    if mode == 'user' :
        gaussMethod.initializationFromUserInput()
    elif mode == 'file' :
        gaussMethod.initializationFromFile()
    gaussMethod.showSystem()
    gaussMethod.applyGaussMethod()
    gaussMethod.showSolutions()

```

1	Вып.	Шохов М.Е.		6.11.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		6.11.19		15
№.		Ф.И.О.	Подп.	Дата		

6. РЕАЛИЗАЦИЯ НА ЯЗЫКЕ FORTRAN

Реализация на языке Fortran

```

module GaussMethod
  integer systemSize
  real, dimension (:,:), allocatable :: coeff
  real, dimension (:), allocatable :: freeTerm
  real, dimension (:), allocatable :: solutions
  character(12) :: filename
  integer isSolutionFounded

  contains
    subroutine init()
      systemSize = 0
      filename = "matrix.txt"
      isSolutionFounded = 0
    end subroutine init

    subroutine destruct()
      deallocate (coeff)
      deallocate (freeTerm)
      deallocate (solutions)
    end subroutine destruct

    subroutine initializationFromUserInput()
      print *, 'Введите количество уравнений: '
      read *, systemSize

      allocate ( coeff(systemSize,systemSize+1) )
      allocate ( freeTerm(systemSize) )
      allocate ( solutions(systemSize) )

      print *, 'Введите коэффициенты:'
      do i = 1, systemSize
        print *, i, ' уравнение:'
        do j = 1, systemSize
          print *, 'a[' , i, '][' , j, ']='
          read *, coeff(i,j)
        end do

        print *, 'Введите свободный член: '
        print *, 'y[' , i, ']= '
        read *, freeTerm(i)
      end do

    end subroutine initializationFromUserInput

    subroutine initializationFromFile()
      real tmp

      open(1,file=filename)

```

1	Вып.	Шохов М.Е.		6.11.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		6.11.19		16
№.		Ф.И.О.	Подп.	Дата		

Реализация на языке Fortran

```

do
    read(1,*, end=1) tmp
    systemSize = systemSize + 1
end do

1 close (1)

allocate ( coeff(systemSize,systemSize+1) )
allocate ( freeTerm(systemSize) )
allocate ( solutions(systemSize) )

open(2,file=filename)

do i = 1,systemSize
    read(2,*) (coeff(i,j),j=1,systemSize+1)
end do

do i = 1,systemSize
    freeTerm(i) = coeff(i,systemSize+1)
    coeff(i,systemSize+1) = 0
end do

close (2)
end subroutine

subroutine showSystem()
write (*,*) 'Введённая система: '
do i = 1, systemSize
    do j = 1, systemSize
        if (coeff(i,j) < 0) then
            write (*,*) '(', coeff(i,j), ')*x', j
        else
            write (*,*) coeff(i,j), '*x', j
        end if
        if (j < systemSize) then
            write (*,*) ' + '
        end if
    end do
    write (*,*) ' = ', freeTerm(i)
end do

end subroutine showSystem

integer function rankOfMatrix(typeOfMatrix)
integer typeOfMatrix
integer :: dimensionOfMatrix = 0
integer :: returnValue = 0
integer :: k, c = 0
real :: tmp = 1

```

1	Вып.	Шохов М.Е.		6.11.19
2	Пров.	Талалушкина Л.В.		6.11.19
№.		Ф.И.О.	Подп.	Дата

Реализация на языке Fortran

```

    if (typeOfMatrix == 0) then
        dimensionOfMatrix = systemSize
    else if (typeOfMatrix == 1) then
        dimensionOfMatrix = systemSize + 1
    end if

    returnValue = 0
    do i = 1,systemSize
        if (coeff(i,i+typeOfMatrix) .NE. 0) then
            returnValue = returnValue + 1
        else
            exit
        end if
    end do

    rankOfMatrix = returnValue
end function rankOfMatrix

subroutine applyGaussMethod()
    real x(systemSize)
    integer rankOfUsualMatrix, rankOfExtendedMatrix
    real c

    do i = 1, systemSize
        coeff(i,systemSize+1) = freeTerm(i)
    end do

    do i = systemSize,2,-1
        if (coeff(i-1,1) < coeff(i,1)) then
            do j = 1,systemSize+1
                c = coeff(i,j)
                coeff(i,j) = coeff(i-1,j)
                coeff(i-1,j) = c
            end do
        end if
    end do

    do i = 1,systemSize
        do j = 1,systemSize+1
            print *, coeff(i,j)
        end do
        print *, "\n"
    end do

    do k = 1,systemSize-1
        do i = k, systemSize-1
            c = (coeff(i+1,k) / coeff(k,k))

```

1	Вып.	Шохов М.Е.		6.11.19
2	Пров.	Талалушкина Л.В.		6.11.19
№.		Ф.И.О.	Подп.	Дата

Реализация на языке Fortran

```

        do j = 1,systemSize+1
            coeff(i+1,j) = coeff(i+1,j) - c*coeff(k,j)
        end do
    end do
end do

rankOfUsualMatrix = rankOfMatrix(0)
rankOfExtendedMatrix = rankOfMatrix(1)

if (rankOfUsualMatrix < rankOfExtendedMatrix) then
    print *, 'Нет решения!'
    stop
else if ( (rankOfUsualMatrix ==
rankOfExtendedMatrix).and.(rankOfUsualMatrix < systemSize) ) then
    print *, 'Система имеет бесконечное множество
решений!'
    stop
else if ( (rankOfUsualMatrix ==
rankOfExtendedMatrix).and.(rankOfUsualMatrix == systemSize) ) then
    print *, 'Система имеет единственное решение!'
end if

do i = systemSize, 1, -1
    !real c
    c = 0
    do j = i,systemSize
        c = c + coeff(i,j) * x(j)
    end do

    x(i) = (coeff(i,systemSize+1) - c) / coeff(i,i)
end do

isSolutionFounded = 1

do i = 1,systemSize
    solutions(i) = x(i)
end do

end subroutine applyGaussMethod

subroutine showSolutions()
    if (isSolutionFounded == 1) then
        print *, 'Решение введённой системы:'
        do i = 1,systemSize
            print *, 'x[', i, ']=', solutions(i)
        end do
    end if
end subroutine showSolutions

```

Реализация на языке Fortran

```
end module
```

```
program main
```

```
    use gaussMethod
```

```
    call init()
```

```
    !call initializationFromUserInput()
```

```
    call initializationFromFile()
```

```
    call showSystem()
```

```
    call applyGaussMethod()
```

```
    call showSolutions()
```

```
    call destruct()
```

```
end
```

1	Вып.	Шохов М.Е.		6.11.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		6.11.19		20
№.		Ф.И.О.	Подп.	Дата		

7. РЕАЛИЗАЦИЯ НА ЯЗЫКЕ MATLAB

Реализация на языке Matlab. Файл: GaussMethod.m

```
classdef GaussMethod < handle
    properties
        systemSize = 0;
        coeff = [];
        freeTerm = [];
        solutions = [];
        filename = "matrix.txt";
        isSolutionFounded = 0;
    end

    methods
        function initializationFromUserInput(obj)
            obj.systemSize = input('Введите количество уравнений: ');

            obj.coeff(obj.systemSize,obj.systemSize+1) = 0;
            obj.freeTerm(obj.systemSize) = 0;
            obj.solutions(obj.systemSize) = 0;

            disp('Введите коэффициенты: ');
            for i = 1:obj.systemSize
                fprintf('%d уравнение: \n', i);
                for j = 1:obj.systemSize
                    fprintf('a[%d][%d]= ', i,j);
                    obj.coeff(i,j) = input(' ');
                end

                disp('Введите свободный член: ');
                fprintf('y[%d]= ',i);
                obj.freeTerm(i) = input(' ');
            end

        end

        function initializationFromFile(obj)
            m = 0;
            n = 0;

            obj.coeff = load(obj.filename);
            [m,n] = size(obj.coeff);

            obj.systemSize = m;
            obj.freeTerm(obj.systemSize) = 0;
            obj.solutions(obj.systemSize) = 0;

            for i = 1:obj.systemSize
                obj.freeTerm(i) = obj.coeff(i,obj.systemSize+1);
                obj.coeff(i,obj.systemSize+1) = 0;
            end
        end
    end
end
```

1	Вып.	Шохов М.Е.		6.11.19
2	Пров.	Талалушкина Л.В.		6.11.19
№.		Ф.И.О.	Подп.	Дата

ЛР по «Численным методам»-НГТУ-(17-ПМ)

Лист

21

Реализация на языке Matlab. Файл: GaussMethod.m

```
function showSystem(obj)
    disp('Введённая система:');
    for i = 1:obj.systemSize
        for j = 1:obj.systemSize
            if (obj.coeff(i,j) < 0)
                fprintf(' (%f)*x%d',obj.coeff(i,j), j);
            else
                fprintf('%f*x%d',obj.coeff(i,j), j);
            end

            if (j < obj.systemSize)
                fprintf(' + ');
            end
        end
        fprintf(' = %d\n', obj.freeTerm(i));
    end
end

function applyGaussMethod(obj)
    x(obj.systemSize) = 0;
    c = 0.0;

    rankOfUsualMatrix = rank(obj.coeff);

    for i = 1: obj.systemSize
        obj.coeff(i,obj.systemSize+1) = obj.freeTerm(i);
    end

    rankOfExtendedMatrix = rank(obj.coeff);

    if (rankOfUsualMatrix < rankOfExtendedMatrix)
        disp ('Нет решения!');
        return
    elseif ( (rankOfUsualMatrix == rankOfExtendedMatrix) &&
(rankOfUsualMatrix < obj.systemSize) )
        disp ('Система имеет бесконечное множество
решений!');
        return
    elseif ( (rankOfUsualMatrix == rankOfExtendedMatrix) &&
(rankOfUsualMatrix == obj.systemSize) )
        disp ('Система имеет единственное решение! ');
    end

    for i = obj.systemSize: -1:2
        if (obj.coeff(i-1,1) < obj.coeff(i,1))
            for j = 1:obj.systemSize+1
                c = obj.coeff(i,j);
                obj.coeff(i,j) = obj.coeff(i-1,j);
                obj.coeff(i-1,j) = c;
            end
        end
    end
end
```

1	Вып.	Шохов М.Е.		6.11.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		6.11.19		22
№.		Ф.И.О.	Подп.	Дата		

Реализация на языке Matlab. Файл: GaussMethod.m

```

        for k = 1:obj.systemSize-1
            for i = k: obj.systemSize-1
                c = (obj.coeff(i+1,k) / obj.coeff(k,k));

                for j = 1:obj.systemSize+1
                    obj.coeff(i+1,j) = obj.coeff(i+1,j) -
c*obj.coeff(k,j);
                end
            end
        end

        for i = obj.systemSize: -1: 1
            c = 0;
            for j = i:obj.systemSize
                c = c + obj.coeff(i,j) * x(j);
            end

            x(i) = (obj.coeff(i,obj.systemSize+1) - c) /
obj.coeff(i,i);
        end

        obj.isSolutionFounded = 1;

        for i = 1:obj.systemSize
            obj.solutions(i) = x(i);
        end

    end

function showSolutions(obj)
    if (obj.isSolutionFounded == 1)
        disp('Решение введённой системы:');
        for i = 1:obj.systemSize
            fprintf ('x[%d]=%f\n',i, obj.solutions(i));
        end
    end
end

end
end
end

```

Реализация на языке Matlab. Файл: main.m

```
function main
    gauss = GaussMethod

    %initializationFromUserInput(gauss)
    initializationFromFile(gauss)
    showSystem(gauss)
    applyGaussMethod(gauss)
    showSolutions(gauss)
end
```

1	Вып.	Шохов М.Е.		6.11.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		6.11.19		24
№.		Ф.И.О.	Подп.	Дата		

8. РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ ПРОГРАММ

Результат программы на языке C++

```

Терминал
Введенная система:
3.81*x0 + 0.25*x1 + 1.28*x2 + 0.75*x3 = 4.21
2.25*x0 + 1.32*x1 + 4.58*x2 + 0.49*x3 = 6.47
5.31*x0 + 6.28*x1 + 0.98*x2 + 1.04*x3 = 2.38
9.39*x0 + 2.45*x1 + 3.35*x2 + 2.28*x3 = 10.48
Система имеет единственное решение!
Решение введенной системы:
x[0]=0.440928
x[1]=-0.430091
x[2]=1.15457
x[3]=1.54631
Для закрытия данного окна нажмите <ВВОД>...

```

Результат программы на языке Python

```

maxwell@maxwell-HP-Laptop-15-bw0xx: ~/numerical method
maxwell@maxwell-HP-Laptop-15-bw0xx:~$ cd numerical\ method/
maxwell@maxwell-HP-Laptop-15-bw0xx:~/numerical method$ python3 gauss.py file
Введенная система:
3.81*x0 + 0.25*x1 + 1.28*x2 + 0.75*x3 = 4.21
2.25*x0 + 1.32*x1 + 4.58*x2 + 0.49*x3 = 6.47
5.31*x0 + 6.28*x1 + 0.98*x2 + 1.04*x3 = 2.38
9.39*x0 + 2.45*x1 + 3.35*x2 + 2.28*x3 = 10.48
Система имеет единственное решение!
Решения введенной системы:
x[0]= 0.4409282857946226
x[1]= -0.4300914420933729
x[2]= 1.154571857143513
x[3]= 1.5463121526695127
maxwell@maxwell-HP-Laptop-15-bw0xx:~/numerical method$ 

```

1	Вып.	Шохов М.Е.		6.11.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист 25
2	Пров.	Талалушкина Л.В.		6.11.19		
№.		Ф.И.О.	Подп.	Дата		

Результат программы на языке Fortran

```

Fgauss
Система имеет единственное решение!
Решение введённой системы:
x[      1 ]=  0.440928817
x[      2 ]= -0.430091441
x[      3 ]=  1.15457189
x[      4 ]=  1.54630971

Process returned 0 (0x0)   execution time : 0.002 s
Press ENTER to continue.
  
```

Результат программы на языке Matlab

```

Command Window
Введённая система:
3.810000*x1 + 0.250000*x2 + 1.280000*x3 + 0.750000*x4 = 4.210000e+00
2.250000*x1 + 1.320000*x2 + 4.580000*x3 + 0.490000*x4 = 6.470000e+00
5.310000*x1 + 6.280000*x2 + 0.980000*x3 + 1.040000*x4 = 2.380000e+00
9.390000*x1 + 2.450000*x2 + 3.350000*x3 + 2.280000*x4 = 1.048000e+01
Система имеет единственное решение!
Решение введённой системы:
x[1]=0.440928
x[2]=-0.430091
x[3]=1.154572
x[4]=1.546312
fx >>
  
```

	C++	Python	Fortran	Matlab
X1	0.440928	0.4409282857946226	0.440928817	0.440928
X2	-0.430091	-0.4300914420933729	-0.430091441	-0.430091
X3	1.15457	1.154571857143513	1.15457189	1.154572
X4	1.54631	1.5463121526695127	1.54630971	1.546312

1	Вып.	Шохов М.Е.		6.11.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		6.11.19		27
№.		Ф.И.О.	Подп.	Дата		

9. ВЫВОД

В ходе выполнения лабораторной работы №1 был изучен и программно реализован метод решения системы линейных алгебраических уравнений – метод Гаусса на языках высокого уровня (C++, Python, Fortran, Matlab).

1	Вып.	Шохов М.Е.		6.11.19	ЛР по «Численным методам»-НГТУ-(17-ПМ)	Лист
2	Пров.	Талалушкина Л.В.		6.11.19		28
№.		Ф.И.О.	Подп.	Дата		