**МИНОБРНАУКИ РОССИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ**

**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №1

дисциплина: Вычислительная математика

тема: «Метод Гаусса»

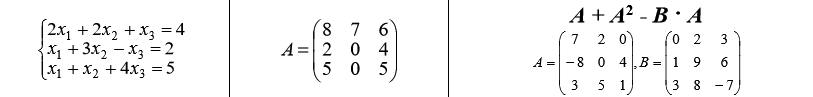
Выполнила: ст. группы ПВ-21

В Браткова Ирина Олеговна

Белгород 2017

**Цель работы**: изучить прямой и обратный ход метода Гаусса для решения систем линейных алгебраических уравнений; изучить схему единственного деления с выбором максимального по модулю элемента; изучить применение метода Гаусса для вычисления определителя матрицы и обратной матрицы; получить практические навыки программной реализации метода Гаусса и решения поставленных задач методом Гаусса с помощью ЭВМ.

**Вариант 4**



**Модуль работы с матрицами**

Заголовочный файл: mod\_matr.h

Файл для работы: mod\_matr.c

**void input\_matr(float\*\*a, size\_t m, size\_t n);** *Назначение:ввод матрицы а размера m на n*

void input\_matr(float \*\*a, size\_t m, size\_t n)

{

int i, j;

for (i=0;i<m;i++)

for (j=0;j<n;j++) scanf("%f",&a[i][j]);

}

**void output\_matr(float\*\*a, size\_t m, size\_t n);**

*Назначение: вывод матрицы а размера m на n*

void output\_matr(float \*\*a, size\_t m, size\_t n)

{

int i, j;

for (i=0;i<m;i++)

{

for (j=0;j<n;j++)

{

printf("%.2f ", a[i][j]);

}

printf("\n");

}

printf("\n");

}

**char summ (float \*\*a, size\_t n, size\_t m, float \*\*b, size\_t k, size\_t l, float \*\*s)**

*Назначение: возвращает 1, если сложение матриц a и b прошло успешно. Возвращает 0, если матрицы не могут быть сложены. Результат сложения записывает в s.*

{ if ((m==k)&&(n==l))

{

for (int i=0; i<m; i++)

for (int j=0; j<n; j++) s[i][j]=a[i][j]+b[i][j];

return 1;

}

else return 0; }

**char razn (float \*\*a, size\_t n, size\_t m, float \*\*b, size\_t k, size\_t l, float \*\*s)**

*Назначение: возвращает 1, если разность матриц a и b найдена успешно, и 0 в противном случае. Результат записывает в s.*

{

if ((m==k)&&(n==l))

{

for (int i=0; i<m; i++)

for (int j=0; j<n; j++) s[i][j]=a[i][j]-b[i][j];

return 1;

}

else return 0;

}

**void um\_al (float \*\*a, size\_t m, size\_t n, float \*\*s)**

*Назначение: умножает матрицу a на число alpha, результат записывает в s.*

{

float al; printf("alpha = ");

scanf ("%f", &al);

for (int i=0; i<m; i++)

{

for (int j=0; j<n; j++)

{

s[i][j] = al\*a[i][j];

}

}

}

**char umn\_matr (float \*\*a, size\_t n, size\_t m, float \*\*b, size\_t k, size\_t l, float \*\*s)**

*Назначение: Возвращает 1, если умножение матрицы a на матрицу b прошло успешно и результат записывает в s. Возвращает 0 в противном случае.*

{

if (n==k)

{

for (int j=0; j<n; j++)

{

for (int i=0; i<m; i++)

{

s[i][j] = 0;

for (int z=0; z<k; z++)

{

s[i][j]=s[i][j]+a[i][z]\*b[z][i];

}

}

}

return 1;

}

else return 0;

}

**void trans (float \*\*a, size\_t m, size\_t n, float \*\*s)**

*Назначение: Транспонирует матрицу a размера m на n, результат записывает в s.*

{

for (int i=0; i<m; i++)

{

for (int j=0; j<n; j++)

{

s[j][i]=a[i][j];

}

}

}

**Модуль работы с матрицами с использованием метода Гаусса**

Заголовочный файл: mod\_gauss.h

Файл для работы: mod\_gauss.c

#include <stdio.h> #include <stdlib.h> #include <locale.h> #include <math.h>

#include "mod\_matr.h" #include "gauss.h" #define EPS 0.0000001

**void str\_swap (float \*\*a, char x, char y)**

*Назначение: обменивает местами строки x, y двумерного массива а.*

{

float \*tmp = a[x]; a[x] = a[y]; a[y] = tmp;

}

**char comp (float x, float y)**

*Назначение: возвращает 1, если x больше y и 0 в противном случае.*

{

if ((x-y)>0) return 1;

else

{

if (fabsf(x-y)<EPS) return 0;

}

}

**char gauss\_nt (float \*\*a, size\_t m, size\_t n)**

*Назначение: прямой ход метода Гаусса. Приводит матрицу к треугольному виду. Нули ниже главной диагонали. Возвращает 1, если приведение прошло успешно.*

{

int i=0, j=0, k=0, y=0; float max=0, mu=0;

for (k=0; k<m; k++)

{

y = k;

max = fabsf(a[k][k]);

for (i=k+1; i<m; i++)

{

if (comp(a[i][k], max))

{

y = i; max = fabsf(a[i][k]);

}

}

if (!comp(max, 0)) return 0;

str\_swap(a, y, k);

for (i=k+1; i<m; i++)

{

mu = a[i][k]/a[k][k];

for (j=0; j<n; j++) a[i][j] -= mu\*a[k][j];

}

}

return 1;

}

**void gauss\_vt (float \*\*a, size\_t m, size\_t n)**

*Назначение: приводит матрицу к треугольному виду с нулями выше главной диагонали с помощью обратного хода метода Гаусса.*

{

for (int k=m-1; k>=0; k--)

{

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

{

float mu = a[i][k]/a[k][k];

for (j=0; j<n; j++) a[i][j] -= mu\*a[k][j];

}

}

}

**char gauss\_slau (float \*\*a, size\_t m, size\_t n)**

*Назначение: вычисляет значение СЛАУ. Возвращает 0, если матрица является невырожденной. Возвращает 1, если поиск решений выполнен успешно. Выводит значения на экран.*

{

int ch = gauss\_nt(a, m, n); if (!ch) return 0;

gauss\_vt (a, m, n);

for (int k=m; k<n; k++)

{

for (int i=m-1; i>=0; i--)

{

float suma=0;

for (int j=m-1; j>i; j--) suma += a[i][j]\*a[j][k];

a[i][k] = (a[i][k]-suma)/a[i][i];

printf("%.2f ", a[i][k]);

}

printf("\n");

}

}

**float gauss\_det (float \*\*a, size\_t m, size\_t n)**

*Назначение: вычисляет значение определителя матрицы через метод Гаусса.*

{

if (!(gauss\_nt(a, m, n))) return 0;

float dett=1;

for (int i=0; i<m; i++) dett \*= a[i][i]; return dett;

}

**char gauss\_obrat (float \*\*a, size\_t m, size\_t n)**

*Назначение: возвращает 1, если обратная матрица была найдена и 0 в противном случае. Выводит полученную обратную матрицу на экран.*

{

if (!gauss\_nt(a, m, n)) return 0;

gauss\_vt(a, m, n);

for (int i = 0; i < m; i++)

{

float num = a[i][i];

for (int j = m ; j < n; j++) a[i][j] /= num; a[i][i] = 1;

}

for (i=0;i<m;i++)

{

for (j=m;j<n;j++)

{

printf("%.2f ", a[i][j]);

}

printf("\n");

}

return 1;

}

**Программа с использованием модуля mod\_gauss.c**

#include <stdio.h> #include <stdlib.h> #include <locale.h>

#include "mod\_matr.h" #include "gauss.h"

void main()

{

setlocale(LC\_ALL,"Rus");

int n, m, i=0, j=0, p, ch;

float \*\*a;

printf("1 - определитель.\n");

printf("2 - обратная матрица.\n");

printf("3 - СЛАУ.\n");

scanf("%d", &ch);

if (ch==1)

{

printf("Размер матрицы: ");

scanf("%d", &m);

a = (float\*\*)calloc(m, sizeof(float\*));

for (i=0; i<m; i++) a[i]=(float\*)calloc(m, sizeof(float));

printf("Введите матрицу: \n");

input\_matr(a, m, m);

printf("\n");

printf("Вы ввели: \n");

output\_matr(a, m, m);

float det = gauss\_det(a, m, m);

printf("det A = %.2f", det);

}

if (ch==2)

{

printf("Размер матрицы: ");

scanf("%d", &m);

n=m+m;

a = (float\*\*)calloc(m, sizeof(float\*));

for (i=0; i<m; i++) a[i]=(float\*)calloc(n, sizeof(float));

printf("Введите матрицу: \n");

input\_matr(a, m, m);

printf("\n");

for (i=0;i<m;i++)

for (j=m; j<n; j++)

{

if ((j-i)==m) a[i][j]=1;

else a[i][j]=0;

}

printf("Вы ввели: \n");

output\_matr(a, m, n);

printf("\n");

printf("Обратная матрица: \n");

ch = gauss\_obrat(a, m, n);

}

if (ch==3)

{

printf("Размер матрицы: ");

scanf("%d", &m);

printf("Правых столбцов: ");

scanf("%d", &p);

n=m+p;

a = (float\*\*)calloc(m, sizeof(float\*));

for (i=0; i<m; i++) a[i]=(float\*)calloc(n, sizeof(float));

printf("Введите матрицу+правые части уравнений: \n");

input\_matr(a, m, n);

printf("Вы ввели: \n");

output\_matr(a, m, n);

printf("\n");

printf("Решение СЛАУ: \n");

gauss\_slau(a, m, n);

}

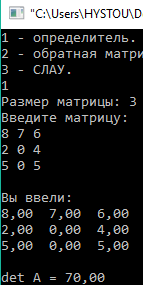
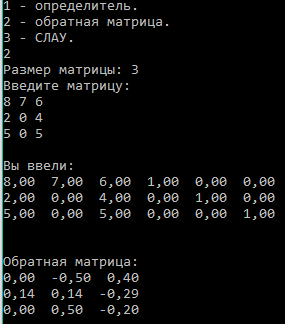
for (i=0;i<m;i++)free(a[i]);

free(a);

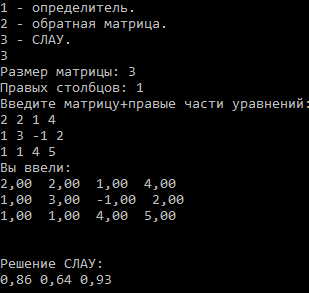
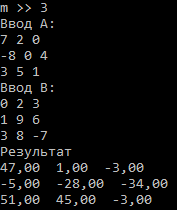
}

**Результат работы программы для задач варианта 4**

*Определитель матрицы: Обратная матрица:*

* *

*СЛАУ: Значение выражения:*

* *