МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

РОССИЙСКОЙ ФЕДЕРАЦИИ

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

Дисциплина: Вычислительная математика

Лабораторная работа №1

Метод Гаусса

|  |  |
| --- | --- |
|  | Выполнила: ст. группы ПВ-21  Зановская А.И.  Проверила: Бондаренко Т.В. |

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 10**

|  |  |  |
| --- | --- | --- |
| Задание | | |
| 1 | 2 | 3 |
|  |  |  |

**Задания к работе**

1. Выполнить вручную действия над матрицами А и В из пункта 3 задания соответствующего варианта.

2. Выполнить следующие действия, не используя метод Гаусса:

1) решить вручную систему линейных алгебраических уравнений с тремя неизвестными из пункта 1 задания соответствующего варианта;

2) найти вручную определитель матрицы А из пункта 2 задания соответствующего варианта;

3) найти вручную матрицу А -1 обратную матрице А из пункта 2 задания соответствующего варианта.

3. Решить вручную методом Гаусса систему линейных алгебраических уравнений с тремя неизвестными из пункта 1 задания соответствующего варианта.

4. Найти вручную с помощью метода Гаусса определитель матрицы А из пункта 2 задания соответствующего варианта.

5. Найти вручную с помощью метода Гаусса матрицу А -1 обратную матрице А из пункта 2 задания соответствующего варианта. Выполнить проверку полученной матрицы на соответствие условию: А·А -1 = Е, где Е ― единичная матрица.

6. Создать модуль для работы с матрицами произвольного порядка, содержащий подпрограммы для умножения двух матриц, умножения числа на матрицу, сложения матриц, вычитания матриц, транспонирования матрицы, умножения матрицы на вектор, ввода и вывода матрицы.

#ifndef MATR\_H\_INCLUDED

#define MATR\_H\_INCLUDED

void input\_matr (int \*\*a, int m, int n)

{

int i, j;

printf ("\nВведите матрицу размера %d\*%d\n", m,n);

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

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

scanf ("%d", &a[i][j]);

}

void input\_vect (int\*v, int n)

{

int i;

printf ("\nВведите вектор размера %d\n",n);

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

scanf ("%d", &v[i]);

}

void output\_matr (int \*\*pa, int m, int n)

{

int i, j;

printf ("\n");

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

{

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

printf ("%d ", pa[i][j]);

printf ("\n");

} printf ("\n");

}

void output\_vect (int\*v, int n)

{

int i;

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

printf ("%d ", v[i]);

printf ("\n");

}

void mult\_n (int \*\*pa, int \*\*pc, int m, int n, int a)

{

int i,j;

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

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

pc[i][j]=pa[i][j]\*a;

return pc;

output\_matr (pa,m,n);

}

void add (int \*\*pa, int \*\*pb, int \*\*pc, int m, int n)

{

int i,j;

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

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

pc[i][j]=pa[i][j]+pb[i][j];

}

void subs (int \*\*pa, int \*\*pb, int \*\*pc, int m, int n)

{

int i,j;

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

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

pc[i][j]=pa[i][j]-pb[i][j];

}

void mult\_matr (int \*\*pa, int \*\*pb, int \*\*pc, int m1, int n1)

{

//m1=n2;n1=m2;

int i, j, x, y;

for (i=0;i<m1;i++)

for (j=0; j<m1;j++)

{

pc[i][j]=0;

for (x=0,y=0;x<n1;x++,y++)

pc[i][j]+=pa[i][x]\*pb[y][j];

}

}

int mult\_v (int \*\*pa, int m, int n, int \*v, int \*va)

{

int i,j,g;

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

{

va[i]=0;

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

va[i]+=pa[i][j]\*v[j];

}

}

void transpose (int \*\*pa, int \*\*pt, int m, int n)

{

int i,j;

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

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

pt[j][i]=pa[i][j];

}

#endif // MATR\_H\_INCLUDED

8. Создать программу для решения следующих задач:

- нахождение методом Гаусса решения системы линейных алгебраических уравнений с одной и той же матрицей коэффициентов при неизвестных и произвольным числом столбцах свободных членов.

Замечание. В прямом ходе метода Гаусса выполняется приведение расширенной матрицы (коэффициенты при неизвестных и свободные члены) к треугольному виду, и одновременно изменяются все столбцы свободных членов. На этапе обратного хода выполняется вычисление решения системы для каждого столбца свободных членов, составляется матрица решений.

- вычисление определителя заданной матрицы методом Гаусса;

- нахождение для заданной матрицы обратной матрицы методом Гаусса.

#include <stdlib.h>

#include <stdio.h>

#include <locale.h>

#include <math.h>

void main()

{

setlocale (LC\_ALL,"Rus");

int n,i,m,k;

float det;

printf ("СЛАУ - 1\nОбратная матрица - 2\n");

scanf ("%d", &k);

if (k==1)

{

printf ("Введите количество переменных: ");

scanf ("%d", &m);

printf ("Введите количество правых частей: ");

scanf("%d",&n);

n+=m;

float \*\*pa;

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

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

pa[i]=(float \*) calloc(n,sizeof(float));

input\_matr (pa,m,n);

output\_matr (pa,m,n);

if(decision (pa,m,n,&det))

{

printf("\ndet = %.2f\n", det);

back\_stroke (pa,m,n);

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

output\_x (pa,m,n);

}

else (printf ("\nСЛАУ имеет несколько решений\n"));

}

else

{

printf ("Введите количество переменных: ");

scanf ("%d", &m);

n=m\*2;

float \*\*pa;

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

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

pa[i]=(float \*) calloc(n,sizeof(float));

input\_matr (pa,m,m);

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

pa[i][i+m]=1;

determination(pa,m,n,&det);

}

}

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

{

int i, j;

printf ("\nВведите матрицу \n");

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

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

scanf("%f", &a[i][j]);

}

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

{

int i, j;

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

{

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

printf("%.4f ", a[i][j]);

printf ("\n");

}

printf ("\n");

}

void subs (float \*\*a, int m, int n,int i)

{

int j,y;

float k;

y=i;

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

{

k=-a[i][y]/a[y][y];

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

a[i][j]+=k\*a[y][j];

}

}

void sub (float \*\*a, int m, int n,int i)

{

int j,y;

float k;

y=i;

for (--i;i>=0;i--)

{

k=-a[i][y]/a[y][y];

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

a[i][j]+=k\*a[y][j];

}

for (j=y+1;j<n;j++)

a[y][j]/=a[y][y];

a[y][y]=1;

}

void swap (float \*\*a, int x, int y, int n)

{

int i; float \*k;

k=a[x];

a[x]=a[y];

a[y]=k;

}

//m - кол-во строк, j - проверяемый столбец

int search (float \*\*a, int m, int j)

{

int i, nom;

float max;

nom = j; max = a[j][j];

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

if (fabs(a[i][j])-fabs(max)>0.001)

{

max=fabs(a[i][j]);

nom=i;

}

return nom;

}

int determination (float \*\*a, int m, int n, float \*det)

{

int i,j=0,k,nom,t=0;

\*det=1;

for (i=0;i<m-1;i++)

{

if (a[i][i]==0)

return 0;

nom=search(a,m,j);

if (nom!=i)

{

swap(a,i,nom,n);

t++;

}

for (k=n;k>i;k--)

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

a[i][i]=1;

subs(a,m,n,i);

j++;

}

if (t%2!=0)

t=1;

else

t=-1;

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

(\*det)\*=a[i][i];

(\*det)\*=t;

printf ("\ndet = %.4f\n",\*det);

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

sub(a,m,n,i);

printf ("\nОбратная матрица\n");

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

{

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

printf("%.4f ", a[i][j]);

printf ("\n");

}

printf ("\n");

}

int decision (float \*\*a, int m, int n, float \*det)

{

int i,j=0,nom,t=0;

\*det=1;

for (i=0;i<m-1;i++)

{

if (a[i][i]==0)

return 0;

nom=search(a,m,j);

if (nom!=i)

{

swap(a,i,nom,n);

t++;

}

subs(a,m,n,i);

j++;

output\_matr(a,m,n);

}

if (t%2!=0)

t=1;

else

t=-1;

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

(\*det)\*=a[i][i];

(\*det)\*=t;

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

{

sub(a,m,n,i);

// j++;

output\_matr(a,m,n);

}

}

void back\_stroke (float \*\*a, int m, int n)

{

int i,j,g,k;

float sum;

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

{

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

{

sum=0;

for(j=m-1;j>i;j--)

sum+=a[i][j]\*a[j][k];

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

output\_matr(a,m,n);

}

}

}

output\_x (float \*\*a, int m, int n)

{

int i,j;

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

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

printf ("\nx[%d] = %.2f", i+1, a[i][j]);

printf ("\n\n");

}

void dec\_high (float \*\*a,float \*\*b, int m, int n)

{

int i,j=0,x,k;

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

{

//x=a[k][k];

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

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

a[k][k]=1;

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

{

for (j=m-1;j>=i;j--)

b[i][j]-=b[k][j]\*a[i][k];

a[i][k]=0;

}

}

}

void mult\_matr (float \*\*pa, float \*\*pb, float \*\*pc, int m1, int n1)

{

//m1=n2;n1=m2;

int i, j, x, y;

for (i=0;i<m1;i++)

for (j=0; j<m1;j++)

{

pc[i][j]=0;

for (x=0,y=0;x<n1;x++,y++)

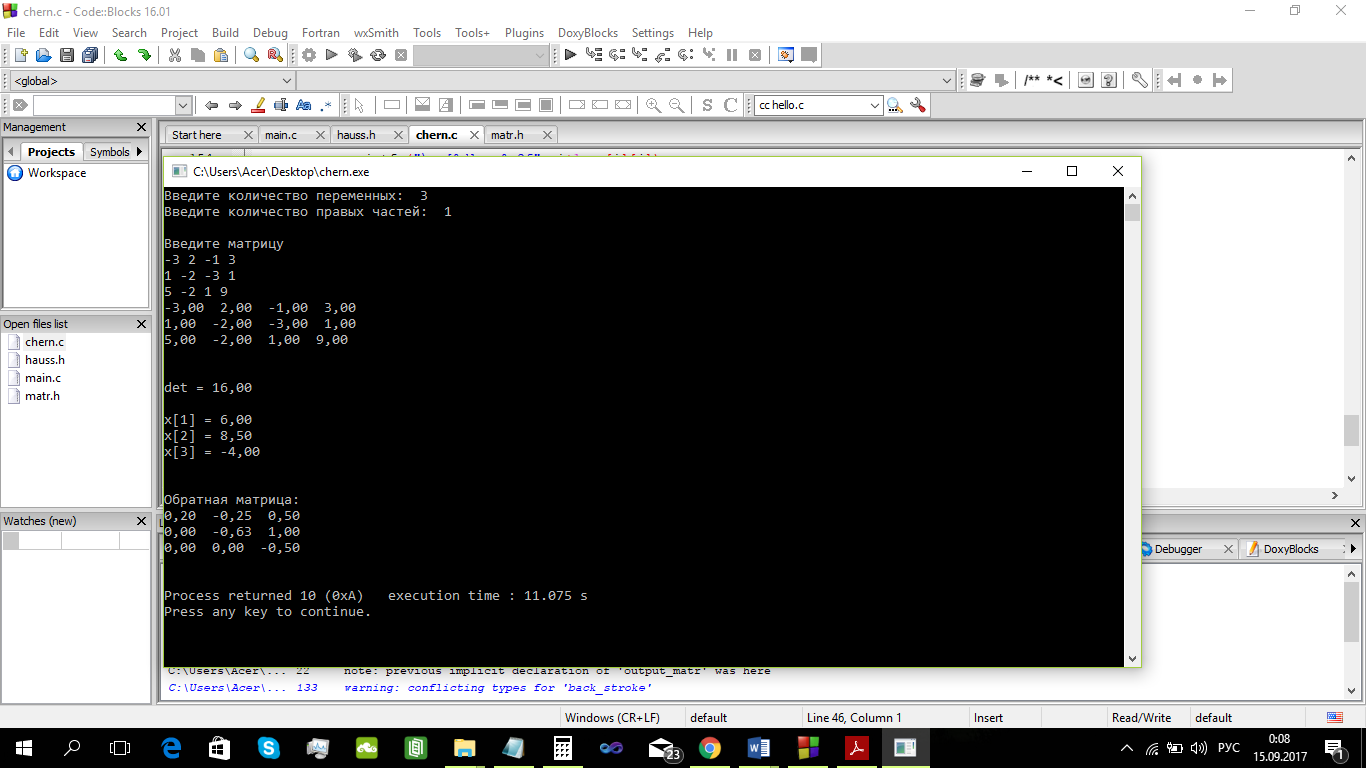
pc[i][j]+=pa[i][x]\*pb[y][j];

}

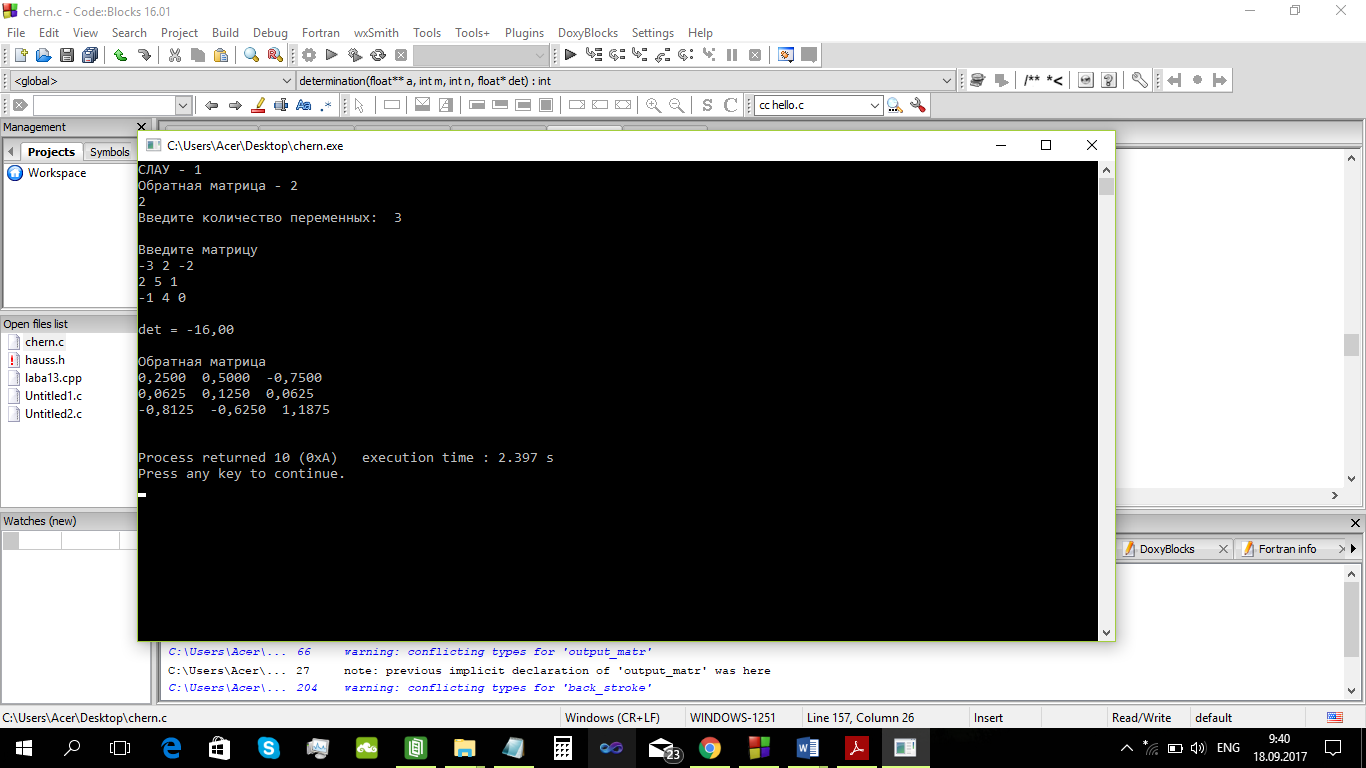
}

9. Решить все задания соответствующего варианта с помощью составленной программы. Выполнить проверку правильности найденного решения системы линейных уравнений и матрицы обратной к заданной матрице с помощью составленной программы, сравнить значения, полученные при решении заданий с помощью метода Гаусса и с использованием произвольного метода решения.

Задание 1.



Задание 2.



Задание 3.

