Лабораторная работа 6

Технологии компьютерного моделирования

Решение систем линейных алгебраических уравнений

Алгоритм выполнения задания

Задания:

1. Ознакомиться с теорией вопроса (представлена в материалах папок:

Теория_1_Классический_метод; Теория_2_Метод оптимального исключения;

Теория_3_Гаусса-Жордана).

2. При выполнении практической части задания руководствуйтесь алгоритмом:

1) Ввести исходные данные и распечатать их;

2) Организовать этап прямого хода и распечатать результат;

3) Организовать этап обратного хода и распечатать результат.

3. Задание 1. Разработать программу по решению СЛУ методом Гаусса (алгоритм исключения неизвестных по столбцам).

В качестве образца используйте Программу из файла Gauss_Klass.jpg

4. Задание 2. Модифицировать программу по решению СЛУ методом Гаусса для реализации алгоритма оптимального исключения неизвестных.

В качестве образца используйте Программу из файла Gauss_Optim_Iskluch.jpg

4. Задание 3. Модифицировать программу по решению СЛУ методом Гаусса для реализации метода Гаусса-Жордана.

В качестве образца используйте Программу из файла Gauss_Jordana.jpg

5. Для выполнения лабораторной работы используйте любой язык программирования.

6. Проверить правильность работы программы на контрольном примере:

A				***B***
5	**7**	**6**	**5**	**23**
7	**10**	**8**	**7**	**32**
6	**8**	**10**	**9**	**33**
5	**7**	**9**	**10**	**31**

7. Измените значения элементов столбца свободных членов на: 0,1 ; 0, 001; 0.01. Например: 23, 1 и т.д.

8. Зафиксируйте результаты в таблице.

9. Сравните полученные результаты с контрольным примером. Поясните их.

Примерные коды для программ:

```
program Gaus;
Uses Crt;
var i,j,n,m,k : integer;
      p,s : real;
      a : array [1..5,1..6] of real;
      x : array [1..5] of real;
begin
      ClrScr;
      { ввод матрицы М1 }
      for i:=1 to 5 do
         begin
           for j:=1 to 6 do
             read (a[i,j]);
             readln;
         end;
       { Печать матрицы а }
       writeln ('Печать матрицы а');
      for i:=1 to 5 do
         begin
           for j:=1 to 6 do
              write (a[i,j]:4:1);
              writeln;
         end;
readln;
clrscr;
n:=5;
{ Процедура прямого хода }
for i:=1 to n-1 do    begin
    p:=a[i,i]; a[i,i]:=1;
      for j:=i+1 to n+1 do begin
         a[i,j]:=a[i,j]/p; end;
      for k:=i+1 to n do
        begin
          for j:=i+1 to n+1 do begin
             a[k,j]:=a[k,j]-a[i,j]*a[k,i]; end;
             a[k,i]:=0;
        end;
    end;
       writeln ('Печать матрицы а');
      for i:=1 to 5 do
         begin
           for j:=1 to 6 do
              write (a[i,j]:4:1);
              writeln;
         end;
{     end;}
     ( writeln ('Печать матрицы а');
      for i:=1 to 5 do
         begin
           for j:=1 to 6 do
              write (a[i,j]:4:1);
              writeln;
         end; )
readln;
{ Обратный ход }
x[n]:=a[n,n+1]/a[n,n];
for i:=n-1 downto 1 do
     begin
         s:=0;
         for j:=i+1 to n do
```

Стандартный вариант

GAUS'MOIN.BAS

Метод оптимального исключения Гаусса

```
DECLARE SUB outMat (m1!, n1!, a1!())
DECLARE SUB ReadMat (m1!, n1!, a1!())
      REM Ввод и вывод исходных данных
      CLS
      n = 5
      DIM a(n + 1, n + 1), x(n)
      DATA 2,4,6,8,2,22
      DATA 3,1,7,4,9,24
      DATA 5,3,8,9,2,27
      DATA 3,1,6,5,4,19
      DATA 6,9,4,1,5,25
      CALL ReadMat(n, n + 1, a())
      CALL outMat(n, n + 1, a())

      REM Процедура прямого хода
      FOR i = 1 TO n - 1
      FOR k = i + 1 TO n
      a(k, i) = a(k, i) / a(i, i)
      FOR j = i + 1 TO n + 1
      a(k, j) = a(k, j) - a(i, j) * a(k, i)
      NEXT j
      a(k, i) = 0
      NEXT k
      CALL outMat(n, n + 1, a())
      INPUT "kontrol "; pppp
      NEXT i
      PRINT : PRINT
      CALL outMat(n, n + 1, a())
      REM Обратный ход
      x(n) = a(n, n + 1) / a(n, n)
      FOR i = n - 1 TO 1 STEP -1
      s = 0
      FOR j = i + 1 TO n
      s = s + a(i, j) * x(j)
      NEXT j
      x(i) = (a(i, n + 1) - s) / a(i, i)
      NEXT i
      REM Корни уравнений
      FOR i = 1 TO n
      PRINT "x="; x(i), i
      NEXT i
      END


   SUB outMat (m1, n1, a1())
   FOR i = 1 TO m1
   FOR j = 1 TO n1
   PRINT a1(i, j);
   NEXT j
   PRINT
   NEXT i
  PRINT


 END SUB

      SUB ReadMat (m1, n1, a1())
      FOR i = 1 TO m1
        FOR j = 1 TO n1
           READ a1(i, j)
        NEXT j
```

```
END SUB
DECLARE SUB outmat (m1!, n1!, a1!())
DECLARE SUB ReadMat (m1!, n1!, a1!())
        REM Ввод и вывод исходных данных
        CLS
        n = 5
        DIM a(n + 1, n + 1), x(n)
        DATA 2,4,6,8,2,22
        DATA 3,1,7,4,9,24
        DATA 5,3,8,9,2,27
        DATA 3,1,6,5,4,19
        DATA 6,9,4,1,5,25
        CALL ReadMat(n, n + 1, a())
        CALL outmat(n, n + 1, a())

        REM Процедура прямого хода
        FOR i = 1 TO n
        FOR k = 1 TO n
            IF k = i THEN GOTO 50
        a(k, i) = a(k, i) / a(i, i)
        FOR j = i + 1 TO n + 1
        a(k, j) = a(k, j) - a(i, i) * a(k, i)
        NEXT j
        a(k, i) = 0
50     NEXT k
        CALL outmat(n, n + 1, a())
        INPUT "kontrol"; pppp
        FOR i = 1 TO n - 1
        p = a(i, i): a(i, i) = 1
        FOR j = i + 1 TO n - 1
        a(i, j) = a(i, j) / p
```

gausjord

```
        NEXT j
        FOR k = i + 1 TO n
        FOR j = i + 1 TO n - 1
        a(k, j) = a(k, j) - a(i, j) * a(k, i)
        NEXT j
        a(k, i) = 0
        NEXT k
        CALL outmat(n, n - 1, a())
        INPUT "kontrol "; pppp
        NEXT i
        PRINT : PRINT
        CALL outmat(n, n - 1, a())
        REM Обратный ход
        x(n) = a(n, n + 1) / a(n, n)
        FOR i = n - n1, a1())
   FOR i = 1 TO m1
   FOR j = 1 TO n1
   PRINT a1(i, j);
   NEXT j
   PRINT
   NEXT i
 PRINT


END SUB

        SUB ReadMat (m1, n1, a1())
        FOR i = 1 TO m1
          FOR j = 1 TO n1
             READ a1(i, j)
          NEXT j
        NEXT i

END SUB
```

gausjord
(конец)

Решение:

Метод Гаусса, модифицированный для реализации алгоритма оптимального исключения неизвестных

```
Введите число уравнений системы: 4
Введите систему:
5
7
6
5
23
7
10
8
7
32
6
8
10
9
33
5
7
9
10
31
Система:
    5.00    7.00    6.00    5.00   23.00
    7.00   10.00    8.00    7.00   32.00
    6.00    8.00   10.00    9.00   33.00
    5.00    7.00    9.00   10.00   31.00
Ответ:
1.000000
1.000000
1.000000
1.000000
> 
```

Другие значения:

```
> clang-7 -pthread -lm -o main mai
> ./main
Введите число уравнений системы: 4
Введите систему:
5
7
6
5
1
7
10
8
7
1
6
8
10
9
1
5
7
9
10
1
Система:
    5.00    7.00    6.00    5.00    1.00
    7.00   10.00    8.00    7.00    1.00
    6.00    8.00   10.00    9.00    1.00
    5.00    7.00    9.00   10.00    1.00
Ответ:
20.000000
-12.000000
-5.000000
3.000000
> 
```

```
Введите число уравнений системы: 4
Введите систему:
5
7
6
5
5
7
10
8
7
7
6
8
10
9
6
5
7
9
10
1
Система:
    5.00     7.00     6.00     5.00     5.00
    7.00    10.00     8.00     7.00     7.00
    6.00     8.00    10.00     9.00     6.00
    5.00     7.00     9.00    10.00     1.00
Ответ:
-39.000000
24.000000
12.000000
-8.000000
>
```

Метод Гаусса-Жордана

```
Enter number of unknowns: 4
Enter coefficients of Augmented Matrix:
a[1][1] = 5
a[1][2] = 7
a[1][3] = 6
a[1][4] = 5
a[1][5] = 23
a[2][1] = 7
a[2][2] = 10
a[2][3] = 8
a[2][4] = 7
a[2][5] = 32
a[3][1] = 6
a[3][2] = 8
a[3][3] = 10
a[3][4] = 9
a[3][5] = 33
a[4][1] = 5
a[4][2] = 7
a[4][3] = 9
a[4][4] = 10
a[4][5] = 31

Solution:
x[1] = 1.000
x[2] = 1.000
x[3] = 1.000
x[4] = 1.000
>
```

Другие значения:

```
Enter number of unknowns: 4
Enter coefficients of Augmented Matrix:
a[1][1] = 5
a[1][2] = 7
a[1][3] = 6
a[1][4] = 5
a[1][5] = 1
a[2][1] = 7
a[2][2] = 10
a[2][3] = 8
a[2][4] = 7
a[2][5] = 1
a[3][1] = 6
a[3][2] = 8
a[3][3] = 10
a[3][4] = 9
a[3][5] = 1
a[4][1] = 5
a[4][2] = 7
a[4][3] = 9
a[4][4] = 10
a[4][5] = 1

Solution:
x[1] = 20.000
x[2] = -12.000
x[3] = -5.000
x[4] = 3.000
```

```
Enter number of unknowns: 4
Enter coefficients of Augmented Matrix:
a[1][1] = 5
a[1][2] = 7
a[1][3] = 6
a[1][4] = 5
a[1][5] = 5
a[2][1] = 7
a[2][2] = 10
a[2][3] = 8
a[2][4] = 7
a[2][5] = 7
a[3][1] = 6
a[3][2] = 8
a[3][3] = 10
a[3][4] = 9
a[3][5] = 6
a[4][1] = 5
a[4][2] = 7
a[4][3] = 9
a[4][4] = 10
a[4][5] = 1

Solution:
x[1] = -39.000
x[2] = 24.000
x[3] = 12.000
x[4] = -8.000
```

Код:

Метод Гаусса, модифицированный для реализации алгоритма оптимального исключения неизвестных

```c
#define N 20
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
void glavelem( int k, double mas[] [N + 1], int n, int otv[] )
{
  int i, j, i_max = k, j_max = k;
  double temp;
  //Ищем максимальный по модулю элемент
  for ( i = k; i < n; i++ )
    for ( j = k; j < n; j++ )
      if ( fabs( mas[i_max] [j_max] ) < fabs( mas[i] [j] ) )
      {
        i_max = i;
        j_max = j;
      }
  //Переставляем строки
  for ( j = k; j < n + 1; j++ )
  {
    temp = mas[k] [j];
    mas[k] [j] = mas[i_max] [j];
    mas[i_max] [j] = temp;
  }
  //Переставляем столбцы
  for ( i = 0; i < n; i++ )
  {
    temp = mas[i] [k];
    mas[i] [k] = mas[i] [j_max];
    mas[i] [j_max] = temp;
  }
  //Учитываем изменение порядка корней
  i = otv[k];
  otv[k] = otv[j_max];
  otv[j_max] = i;
}
int main( void )
{
  double mas[N] [N + 1];
  double x[N]; //Корни системы
  int otv[N]; //Отвечает за порядок корней
  int i, j, k, n;
  //Ввод данных
  do
  {
    printf( "Введите число уравнений системы: " );
    scanf( "%d", & n );
    if ( N < n )
      printf( "Слишком большое число уравнений. Повторите ввод\n" );
  }
```

```c
    while ( N < n );
    printf( "Введите систему:\n" );
    for ( i = 0; i < n; i++ )
        for ( j = 0; j < n + 1; j++ )
            scanf( "%lf", & mas[i] [j] );
    //Вывод введенной системы
    printf( "Система:\n" );
    for ( i = 0; i < n; i++ )
    {
        for ( j = 0; j < n + 1; j++ )
            printf( "%7.2f ", mas[i] [j] );
        printf( "\n" );
    }
    //Сначала все корни по порядку
    for ( i = 0; i < n + 1; i++ )
        otv[i] = i;
    //Прямой ход метода Гаусса
    for ( k = 0; k < n; k++ )
    { //На какой позиции должен стоять главный элемент
        glavelem( k, mas, n, otv ); //Установка главного элемента
        if ( fabs( mas[k] [k] ) < 0.0001 )
        {
            printf( "Система не имеет единственного решения" );
            return ( 0 );
        }
        for ( j = n; j >= k; j-- )
            mas[k] [j] /= mas[k] [k];
        for ( i = k + 1; i < n; i++ )
            for ( j = n; j >= k; j-- )
                mas[i] [j] -= mas[k] [j] * mas[i] [k];
    }
    //Обратный ход
    for ( i = 0; i < n; i++ )
        x[i] = mas[i] [n];
    for ( i = n - 2; i >= 0; i-- )
        for ( j = i + 1; j < n; j++ )
            x[i] -= x[j] * mas[i] [j];
    //Вывод результата
    printf( "Ответ:\n" );
    for ( i = 0; i < n; i++ )
        for ( j = 0; j < n; j++ )
            if ( i == otv[j] )
            { //Расставляем корни по порядку
                printf( "%f\n", x[j] );
                break;
            }
    return ( 0 );
}
```

Метод Гаусса-Жордана

```c
#include<stdio.h>
#include<math.h>

#define SIZE 10

int main()
{
        float a[SIZE][SIZE], x[SIZE], ratio;
        int i,j,k,n;
        printf("Enter number of unknowns: ");
        scanf("%d", &n);
        printf("Enter coefficients of Augmented Matrix:\n");
        for(i=1;i<=n;i++)
        {
            for(j=1;j<=n+1;j++)
            {
                printf("a[%d][%d] = ",i,j);
                scanf("%f", &a[i][j]);
            }
        }
        for(i=1;i<=n;i++)
        {
            if(a[i][i] == 0.0)
            {
                printf("Mathematical Error!");
                return(1);
            }
            for(j=1;j<=n;j++)
            {
                if(i!=j)
                {
                    ratio = a[j][i]/a[i][i];
                    for(k=1;k<=n+1;k++)
                    {
                        a[j][k] = a[j][k] - ratio*a[i][k];
                    }
                }
            }
        }
        /* Reverse */
        /*for(i=n-1;-1;i--)
        {
          ratio = a[i][i];
          for (int l=0; i;)
          {
              for (int k=4; i-1; i--)
              {
                a[l][k] -=  a[i][k] * a[l][i] / ratio;
              }
          }
          a[i][i] /= ratio;
          for (int x=4;5;)
              a[i][x] /= ratio;
        }*/


        for(i=1;i<=n;i++)
        {
            x[i] = a[i][n+1]/a[i][i];
        }

        printf("\nSolution:\n");
        for(i=1;i<=n;i++)
        {
            printf("x[%d] = %0.3f\n",i, x[i]);
        }
        scanf("");
        return(0);
}
```

Вывод:

При изменении значений свободных переменных меняется значения решения. Это происходит из-за того, что задавая свободным переменным иные значения, можно получить иные частные решения данной системы. Подобных решений бесконечное множество.

В итоге, нам удалось выполнить поставленную задачу, разработать программы, провести исследование, и изучить решение систем линейных алгебраических уравнений.