Министерство образования и науки РФ
Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа программной инженерии

Отчёт по лабораторной работе №2

по дисциплине «Вычислительная математика»

Выполнил студент гр. 3530904/90002 Мэн Цзянин

Преподаватель

С.П.Воскобойников

Санкт-Петербург
2021

Оглавление

Постановка задачи

В А Р И А Н Т N 15

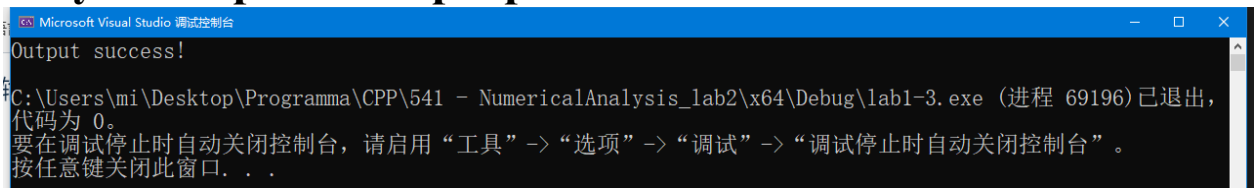
Сравнить два вектора: $x_1=A^{-1}b$ и x_2, полученный непосредственным решением системы с использованием программ DECOMP и SOLVE. Обратную матрицу A^{-1} вычислить с помощью DECOMP и SOLVE. Система $Ax=b$ зависит от параметра p $(p= 1.0, 0.1, 0.01, 0.0001, 0.000001)$. Проанализировать связь числа обусловленности cond и величины : $\delta=\|x_1-x_2\|/\|x_1\|$.

$$\begin{pmatrix} p+27 & -6 & -1 & -6 & -3 & -4 & -3 & -4 \\ -6 & 35 & -1 & -6 & -5 & -6 & -3 & -8 \\ -1 & -1 & 19 & -6 & -8 & -2 & 0 & -1 \\ -6 & -6 & -6 & 36 & -4 & -3 & -4 & -7 \\ -3 & -5 & -8 & -4 & 25 & 0 & -1 & -4 \\ -4 & -6 & -2 & -3 & 0 & 28 & -8 & -5 \\ -3 & -3 & 0 & -4 & -1 & -8 & 21 & -2 \\ -4 & -8 & -1 & -7 & -4 & -5 & -2 & 31 \end{pmatrix} \begin{pmatrix} x^1 \\ x^2 \\ x^3 \\ x^4 \\ x^5 \\ x^6 \\ x^7 \\ x^8 \end{pmatrix} = \begin{pmatrix} 8p+140 \\ -91 \\ -7 \\ 142 \\ 7 \\ -99 \\ 25 \\ -117 \end{pmatrix}$$

Ход работы

1) Построены матрицы A и векторы b в зависимости от изменения параметра p.
2) Найдена обратная матрийа A^-1 с помощью программы DECOMP и SOLVE, получино число обратной матрицы cond(A^-1)
3) Получино решение x1 = A^-1*b
4) Для матрицы P(P==A) найдено её LU разложение с помощью программы DECOMP, а затем необходимое число раз (а именно N = 8) решены системы с помощью программы SOLVE. Получино решение x2
5) Вычислено вектор невязки |x1-x2|
6) Вычислена величина $\delta=\|x_1-x_2\|/\|x_1\|$
7) Получено число обусловленности
8) Исследована связь числа обусловленности и delta.

Результат работы программы



result-lab2-table.xlsx

связь числа обусловленности cond и величины delta=||x1-x2||/||x1||

p	cond(A^-1)	delta
1	5.93E+02	1.75E-14
0.1	5777	1.22E-13
0.01	5.76E+04	1.27E-12
0.0001	5.76E+06	1.47E-10
0.000001	5.76E+08	1.14E-08

талица 1

p1=1							
matrix A							
2.80E+01	-6.00E+00	-1.00E+00	-6.00E+00	-3.00E+00	-4.00E+00	-3.00E+00	-4.00E+00
-6.00E+00	3.50E+01	-1.00E+00	-6.00E+00	-5.00E+00	-6.00E+00	-3.00E+00	-8.00E+00
-1.00E+00	-1.00E+00	1.90E+01	-6.00E+00	-8.00E+00	-2.00E+00	0.00E+00	-1.00E+00
-6.00E+00	-6.00E+00	-6.00E+00	3.60E+01	-4.00E+00	-3.00E+00	-4.00E+00	-7.00E+00
-3.00E+00	-5.00E+00	-8.00E+00	-4.00E+00	2.50E+01	0.00E+00	-1.00E+00	-4.00E+00
-4.00E+00	-6.00E+00	-2.00E+00	-3.00E+00	0.00E+00	2.80E+01	-8.00E+00	-5.00E+00
-3.00E+00	-3.00E+00	0.00E+00	-4.00E+00	-1.00E+00	-8.00E+00	2.10E+01	-2.00E+00
-4.00E+00	-8.00E+00	-1.00E+00	-7.00E+00	-4.00E+00	-5.00E+00	-2.00E+00	3.10E+01
matrix A^-1							
1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
1.00E+00	1.06E+00	1.03E+00	1.03E+00	1.03E+00	1.03E+00	1.03E+00	1.03E+00
1.00E+00	1.03E+00	1.09E+00	1.04E+00	1.05E+00	1.03E+00	1.03E+00	1.03E+00
1.00E+00	1.03E+00	1.04E+00	1.06E+00	1.03E+00	1.03E+00	1.03E+00	1.03E+00
1.00E+00	1.03E+00	1.05E+00	1.03E+00	1.07E+00	1.03E+00	1.03E+00	1.03E+00
1.00E+00	1.03E+00	1.03E+00	1.03E+00	1.03E+00	1.07E+00	1.04E+00	1.03E+00
1.00E+00	1.03E+00	1.03E+00	1.03E+00	1.03E+00	1.04E+00	1.08E+00	1.03E+00
1.00E+00	1.03E+00	1.03E+00	1.03E+00	1.03E+00	1.03E+00	1.03E+00	1.06E+00
cond(A^-1)=5.929e+02							
x1(invA1*b1)=	x2=	|x1-x2|=					
8.00E+00	8.00E+00	4.89E-14					
1.00E+00	1.00E+00	3.65E-14					
4.00E+00	4.00E+00	9.37E-14					
7.00E+00	7.00E+00	3.55E-14					
4.00E+00	4.00E+00	9.28E-14					
7.11E-14	-2.31E-14	9.41E-14					
4.00E+00	4.00E+00	6.53E-14					
0.00E+00	-2.26E-14	2.26E-14					
||x2-x1||/||x1||=1.748e-14							

p2=0.1							
matrix A							
2.71E+01	-6.00E+00	-1.00E+00	-6.00E+00	-3.00E+00	-4.00E+00	-3.00E+00	-4.00E+00
-6.00E+00	3.50E+01	-1.00E+00	-6.00E+00	-5.00E+00	-6.00E+00	-3.00E+00	-8.00E+00
-1.00E+00	-1.00E+00	1.90E+01	-6.00E+00	-8.00E+00	-2.00E+00	0.00E+00	-1.00E+00
-6.00E+00	-6.00E+00	-6.00E+00	3.60E+01	-4.00E+00	-3.00E+00	-4.00E+00	-7.00E+00
-3.00E+00	-5.00E+00	-8.00E+00	-4.00E+00	2.50E+01	0.00E+00	-1.00E+00	-4.00E+00
-4.00E+00	-6.00E+00	-2.00E+00	-3.00E+00	0.00E+00	2.80E+01	-8.00E+00	-5.00E+00
-3.00E+00	-3.00E+00	0.00E+00	-4.00E+00	-1.00E+00	-8.00E+00	2.10E+01	-2.00E+00
-4.00E+00	-8.00E+00	-1.00E+00	-7.00E+00	-4.00E+00	-5.00E+00	-2.00E+00	3.10E+01
matrix A^-1							
1.00E+01	1.00E+01	1.00E+01	1.00E+01	1.00E+01	1.00E+01	1.00E+01	1.00E+01
1.00E+01	1.01E+01	1.00E+01	1.00E+01	1.00E+01	1.00E+01	1.00E+01	1.00E+01
1.00E+01	1.00E+01	1.01E+01	1.00E+01	1.01E+01	1.00E+01	1.00E+01	1.00E+01
1.00E+01	1.00E+01	1.00E+01	1.01E+01	1.00E+01	1.00E+01	1.00E+01	1.00E+01
1.00E+01	1.00E+01	1.01E+01	1.00E+01	1.01E+01	1.00E+01	1.00E+01	1.00E+01
1.00E+01	1.00E+01	1.00E+01	1.00E+01	1.00E+01	1.01E+01	1.00E+01	1.00E+01
1.00E+01	1.00E+01	1.00E+01	1.00E+01	1.00E+01	1.00E+01	1.01E+01	1.00E+01
1.00E+01	1.00E+01	1.00E+01	1.00E+01	1.00E+01	1.00E+01	1.00E+01	1.01E+01
cond(A^-1)=5.777e+03							
(invA1*b1	x2=	|x1-x2|=					
8.00E+00	8.00E+00	3.13E-13					
1.00E+00	1.00E+00	5.40E-13					
4.00E+00	4.00E+00	5.39E-13					
7.00E+00	7.00E+00	7.68E-13					
4.00E+00	4.00E+00	5.39E-13					
4.55E-13	1.43E-13	3.12E-13					
4.00E+00	4.00E+00	8.44E-14					
4.55E-13	1.43E-13	3.12E-13					
||x2-x1||/||x1||=1.217e-13							

p3=0.01

matrix A

2.70E+01	-6.00E+00	-1.00E+00	-6.00E+00	-3.00E+00	-4.00E+00	-3.00E+00	-4.00E+00
-6.00E+00	3.50E+01	-1.00E+00	-6.00E+00	-5.00E+00	-6.00E+00	-3.00E+00	-8.00E+00
-1.00E+00	-1.00E+00	1.90E+01	-6.00E+00	-8.00E+00	-2.00E+00	0.00E+00	-1.00E+00
-6.00E+00	-6.00E+00	-6.00E+00	3.60E+01	-4.00E+00	-3.00E+00	-4.00E+00	-7.00E+00
-3.00E+00	-5.00E+00	-8.00E+00	-4.00E+00	2.50E+01	0.00E+00	-1.00E+00	-4.00E+00
-4.00E+00	-6.00E+00	-2.00E+00	-3.00E+00	0.00E+00	2.80E+01	-8.00E+00	-5.00E+00
-3.00E+00	-3.00E+00	0.00E+00	-4.00E+00	-1.00E+00	-8.00E+00	2.10E+01	-2.00E+00
-4.00E+00	-8.00E+00	-1.00E+00	-7.00E+00	-4.00E+00	-5.00E+00	-2.00E+00	3.10E+01

matrix A^-1

1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02
1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02
1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02
1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02
1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02
1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02
1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02
1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02

cond(A^-1)=5.762e+04

.(invA1*b1	x2=	|x1-x2|=
8.00E+00	8.00E+00	4.89E-12
1.00E+00	1.00E+00	3.07E-12
4.00E+00	4.00E+00	1.04E-11
7.00E+00	7.00E+00	1.26E-12
4.00E+00	4.00E+00	6.71E-12
7.28E-12	8.53E-12	1.26E-12
4.00E+00	4.00E+00	4.89E-12
5.46E-12	8.53E-12	3.08E-12

||x2-x1||/||x1||=1.268e-12

p4=0.0001

matrix A

2.70E+01	-6.00E+00	-1.00E+00	-6.00E+00	-3.00E+00	-4.00E+00	-3.00E+00	-4.00E+00
-6.00E+00	3.50E+01	-1.00E+00	-6.00E+00	-5.00E+00	-6.00E+00	-3.00E+00	-8.00E+00
-1.00E+00	-1.00E+00	1.90E+01	-6.00E+00	-8.00E+00	-2.00E+00	0.00E+00	-1.00E+00
-6.00E+00	-6.00E+00	-6.00E+00	3.60E+01	-4.00E+00	-3.00E+00	-4.00E+00	-7.00E+00
-3.00E+00	-5.00E+00	-8.00E+00	-4.00E+00	2.50E+01	0.00E+00	-1.00E+00	-4.00E+00
-4.00E+00	-6.00E+00	-2.00E+00	-3.00E+00	0.00E+00	2.80E+01	-8.00E+00	-5.00E+00
-3.00E+00	-3.00E+00	0.00E+00	-4.00E+00	-1.00E+00	-8.00E+00	2.10E+01	-2.00E+00
-4.00E+00	-8.00E+00	-1.00E+00	-7.00E+00	-4.00E+00	-5.00E+00	-2.00E+00	3.10E+01

matrix A^-1

1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04
1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04
1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04
1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04
1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04
1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04
1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04
1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04	1.00E+04

cond(A^-1)=5.760e+06

.(invA1*b1	x2=	|x1-x2|=
8.00E+00	8.00E+00	4.26E-10
1.00E+00	1.00E+00	4.26E-10
4.00E+00	4.00E+00	1.94E-10
7.00E+00	7.00E+00	6.59E-10
4.00E+00	4.00E+00	4.26E-10
0.00E+00	4.26E-10	4.26E-10
4.00E+00	4.00E+00	6.59E-10
-4.66E-10	4.26E-10	8.92E-10

||x2-x1||/||x1||=1.468e-10

p1=0.000001							
matrix A							
2.70E+01	-6.00E+00	-1.00E+00	-6.00E+00	-3.00E+00	-4.00E+00	-3.00E+00	-4.00E+00
-6.00E+00	3.50E+01	-1.00E+00	-6.00E+00	-5.00E+00	-6.00E+00	-3.00E+00	-8.00E+00
-1.00E+00	-1.00E+00	1.90E+01	-6.00E+00	-8.00E+00	-2.00E+00	0.00E+00	-1.00E+00
-6.00E+00	-6.00E+00	-6.00E+00	3.60E+01	-4.00E+00	-3.00E+00	-4.00E+00	-7.00E+00
-3.00E+00	-5.00E+00	-8.00E+00	-4.00E+00	2.50E+01	0.00E+00	-1.00E+00	-4.00E+00
-4.00E+00	-6.00E+00	-2.00E+00	-3.00E+00	0.00E+00	2.80E+01	-8.00E+00	-5.00E+00
-3.00E+00	-3.00E+00	0.00E+00	-4.00E+00	-1.00E+00	-8.00E+00	2.10E+01	-2.00E+00
-4.00E+00	-8.00E+00	-1.00E+00	-7.00E+00	-4.00E+00	-5.00E+00	-2.00E+00	3.10E+01
matrix A^-1							
1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06
1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06
1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06
1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06
1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06
1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06
1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06
1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06	1.00E+06
cond(A^-1)=5.760e+08							
.(invA1*b1	x2=	|x1-x2|=					
8.00E+00	8.00E+00	8.73E-08					
1.00E+00	1.00E+00	2.07E-09					
4.00E+00	4.00E+00	4.26E-08					
7.00E+00	7.00E+00	3.19E-08					
4.00E+00	4.00E+00	5.75E-08					
2.98E-08	4.26E-08	1.28E-08					
4.00E+00	4.00E+00	4.26E-08					
0.00E+00	4.26E-08	4.26E-08					
||x2-x1||/||x1||=1.141e-08							

Код программы

```
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <fstream>
#include <iostream>
#include "cmath.h"
#include <iomanip>
#include <vector>

const int DIM_8 = 8;
double cond;
static int flag = -1;
int pivot[DIM_8];
std::ofstream out("lab2-result.txt");

double* minus(double* x1, double* x2)
{
 double* result = new double[DIM_8];
 for (int j = 0; j < 8; j++)
 {
  result[j] = fabs(x1[j] - x2[j]);
 }
 return result;
}

double get_relative_norm(double* x1, double* x2)
{
 double minus_norm = 0;
 double norm_x1 = 0;
 for (int j = 0; j < 8; j++)
 {
  minus_norm += fabs(x1[j] - x2[j]);
  norm_x1 += fabs(x1[j]);
 }
 return (minus_norm / norm_x1);
}


double* set_matrix_A(double p)
{

 double* retP = new double[64];

 retP[0] = p + 27; retP[1] = -6;  retP[2] = -1;  retP[3] = -6;  retP[4] = -3;  retP[5] = -
4;  retP[6] = -3;  retP[7] = -4;
```

```
 retP[8] = -6;     retP[9] = 35;  retP[10] = -1; retP[11] = -6; retP[12] = -5; retP[13] = -6; retP[14] = -3; retP[15] = -8;
 retP[16] = -1;    retP[17] = -1; retP[18] = 19; retP[19] = -6; retP[20] = -8; retP[21] = -2; retP[22] = 0;  retP[23] = -1;
 retP[24] = -6;    retP[25] = -6; retP[26] = -6; retP[27] = 36; retP[28] = -4; retP[29] = -3; retP[30] = -4; retP[31] = -7;
 retP[32] = -3;    retP[33] = -5; retP[34] = -8; retP[35] = -4; retP[36] = 25; retP[37] = 0;  retP[38] = -1; retP[39] = -4;
 retP[40] = -4;    retP[41] = -6; retP[42] = -2; retP[43] = -3; retP[44] = 0;  retP[45] = 28; retP[46] = -8; retP[47] = -5;
 retP[48] = -3;    retP[49] = -3; retP[50] = 0;  retP[51] = -4; retP[52] = -1; retP[53] = -8; retP[54] = 21; retP[55] = -2;
 retP[56] = -4;    retP[57] = -8; retP[58] = -1; retP[59] = -7; retP[60] = -4; retP[61] = -5; retP[62] = -2; retP[63] = 31;

 return retP;
}

double* set_vector_b(double p)
{
 double* ret_b = new double[DIM_8];
 ret_b[0] = 8 * p + 140;
 ret_b[1] = -91;
 ret_b[2] = -7;
 ret_b[3] = 142;
 ret_b[4] = 7;
 ret_b[5] = -99;
 ret_b[6] = 25;
 ret_b[7] = -117;

 return ret_b;

}

void show_matrix(double* arr)
{
 for (int i = 0; i < 8; i++) {
  for (int j = 0; j < 8; j++)
  {
   out << std::scientific << std::setprecision(3) << setiosflags(std::ios::left);
   //out << std::setw(15) << arr[i * 8 + j];
   out << arr[i * 8 + j] << '\t';
  }
  out << std::endl;
 }
 out << std::endl;
}

```

```
86. void show_2d_array(double** arr)
87. {
88.  out << " before decomp, P1=\n";
89.  for (int i = 0; i < 8; i++) {
90.   for (int j = 0; j < 8; j++)
91.   {
92.    out << std::scientific << std::setprecision(3) << setiosflags(std::ios::left);
93.    out << std::setw(15) << arr[i][j];
94.   }
95.   out << std::endl;
96.  }
97.  out << std::endl;
98. }
99.
100.void show_vector(double* v)
101.{
102. for (int i = 0; i < 8; i++)
103. {
104.  out << std::scientific << std::setprecision(3) << setiosflags(std::ios::left);
105.  // out << std::setw(15) << v[i];
106.  out << v[i] << '\n';
107. }
108. out << std::endl;
109.}
110.
111.double* inv(double* arr)
112.{
113. static int flag;
114. int pivot[DIM_8];
115. double* a_inv = new double[64];
116. double b[8] = { 0, 0, 0, 0, 0, 0, 0, 0 };
117.
118. decomp(DIM_8, DIM_8, arr, &cond, pivot, &flag);
119.
120. if (flag != 0)
121. {
122.  out << "Fail to decompose!! flag =" << flag << "\n";
123.  return NULL;
124. }
125.
126. for (int i = 0; i < DIM_8; i++)
127. {
128.  b[i] = 1;
129.  solve(8, 8, arr, b, pivot);
130.  for (int j = 0; j < DIM_8; j++)
131.  {
132.   a_inv[j * 8 + i] = b[j];
133.   b[j] = 0;
```

```
  }
 }

 out << "A^-1:\n";
 show_matrix(a_inv);

 out << "cond(A^-1)=" << cond << "\n\n";

 return a_inv;
}

// matrix multiply by vector
double* multiply_by_vector(double* matrix, double* vec)
{
 double* result_vector = new double[DIM_8];

 double temp_element = 0;
 for (int i = 0; i < DIM_8; ++i)
 {
  temp_element = 0;
  for (int j = 0; j < DIM_8; ++j)
  {
   temp_element += matrix[i * DIM_8 + j] * vec[j];
  }
  result_vector[i] = temp_element;
 }

 return result_vector;
}

double* decomp_solve_2d(double matrix_A[][8], double* vec_b)
{

 decomp(DIM_8, DIM_8, *matrix_A, &cond, pivot, &flag);
 if (flag != 0)
 {
  out << "fail to decompose! flag=" << flag << '\n';
 }
 solve(8, 8, *matrix_A, vec_b, pivot);

 out << "x2(solution with decomp and solve)=\n";
 show_vector(vec_b);

 return vec_b;
}


void lab2()
```

```
{

 using std::endl;
 double cond = 0;

 double p1 = 1.0;
 double p2 = 0.1;
 double p3 = 0.01;
 double p4 = 0.0001;
 double p5 = 0.000001;

 double* b1 = set_vector_b(p1);
 double* b2 = set_vector_b(p2);
 double* b3 = set_vector_b(p3);
 double* b4 = set_vector_b(p4);
 double* b5 = set_vector_b(p5);

 double* A1 = set_matrix_A(p1);
 double* A2 = set_matrix_A(p2);
 double* A3 = set_matrix_A(p3);
 double* A4 = set_matrix_A(p4);
 double* A5 = set_matrix_A(p5);

 double P1[DIM_8][DIM_8] = {
 {p1 + 27, -6, -1, -6, -3,  -4, -3, -4},
 {-6,   35, -1, -6, -5,  -6, -3, -8,},
 {-1,   -1, 19, -6, -8,  -2,  0, -1,},
 {-6,   -6, -6, 36, -4,  -3, -4, -7,},
 {-3,   -5, -8, -4, 25,   0, -1, -4,},
 {-4,   -6, -2, -3,  0,  28,  -8, -5,},
 {-3,   -3,  0, -4, -1,  -8, 21, -2,},
 {-4,   -8, -1, -7, -4,  -5, -2, 31}
 };

 double P2[DIM_8][DIM_8] = {
  {p2 + 27, -6, -1, -6, -3,  -4, -3, -4},
  {-6,   35, -1, -6, -5,  -6, -3, -8,},
  {-1,   -1, 19, -6, -8,  -2,  0, -1,},
  {-6,   -6, -6, 36, -4,  -3, -4, -7,},
  {-3,   -5, -8, -4, 25,   0, -1, -4,},
  {-4,   -6, -2, -3,  0,  28,  -8, -5,},
  {-3,   -3,  0, -4, -1,  -8, 21, -2,},
  {-4,   -8, -1, -7, -4,  -5, -2, 31}
 };

 double P3[DIM_8][DIM_8] = {
  {p3 + 27, -6, -1, -6, -3,  -4, -3, -4},
  {-6,   35, -1, -6, -5,  -6, -3, -8,},
```

```
 {-1,   -1, 19, -6, -8,  -2,  0, -1,},
 {-6,   -6, -6, 36, -4,  -3, -4, -7,},
 {-3,   -5, -8, -4, 25,   0, -1, -4,},
 {-4,   -6, -2, -3,  0,  28,  -8, -5,},
 {-3,   -3,  0, -4, -1,  -8, 21, -2,},
 {-4,   -8, -1, -7, -4,  -5, -2, 31}
};

double P4[DIM_8][DIM_8] = {
 {p4 + 27, -6, -1, -6, -3,  -4, -3, -4},
 {-6,   35, -1, -6, -5,  -6, -3, -8,},
 {-1,   -1, 19, -6, -8,  -2,  0, -1,},
 {-6,   -6, -6, 36, -4,  -3, -4, -7,},
 {-3,   -5, -8, -4, 25,   0, -1, -4,},
 {-4,   -6, -2, -3,  0,  28,  -8, -5,},
 {-3,   -3,  0, -4, -1,  -8, 21, -2,},
 {-4,   -8, -1, -7, -4,  -5, -2, 31}
};

double P5[DIM_8][DIM_8] = {
 {p5 + 27, -6, -1, -6, -3,  -4, -3, -4},
 {-6,   35, -1, -6, -5,  -6, -3, -8,},
 {-1,   -1, 19, -6, -8,  -2,  0, -1,},
 {-6,   -6, -6, 36, -4,  -3, -4, -7,},
 {-3,   -5, -8, -4, 25,   0, -1, -4,},
 {-4,   -6, -2, -3,  0,  28,  -8, -5,},
 {-3,   -3,  0, -4, -1,  -8, 21, -2,},
 {-4,   -8, -1, -7, -4,  -5, -2, 31}
};


// output result
out << "---------------------------- A1 * x = b1 ---------------------------------\n";
out << "p1=" << p1 << "\n\n";
out << "A1:\n";
show_matrix(A1);

double* invA1 = inv(A1);
double* x1_1 = multiply_by_vector(invA1, b1);

out << "x1(invA1*b1)=\n";
show_vector(x1_1);

double* x2_1 = decomp_solve_2d(P1, b1);

out << "|x1-x2|=\n";
show_vector(minus(x1_1, x2_1));

```

```
278. out << "||x2-x1||/||x1||=" << get_relative_norm(x1_1, x2_1) << '\n';
279.
280.
281. out << "---------------------------- A2 * x = b2 ---------------------------------\n";
282. out << "p2=" << p2 << "\n\n";
283.
284. out << "A2:\n";
285. show_matrix(A2);
286.
287. double* invA2 = inv(A2);
288. double* x1_2 = multiply_by_vector(invA2, b2);
289.
290. out << "x1(invA2*b2)=\n";
291. show_vector(x1_2);
292.
293. double* x2_2 = decomp_solve_2d(P2, b2);
294. out << "|x1-x2|=\n";
295. show_vector(minus(x1_2, x2_2));
296.
297.
298. out << "||x2-x1||/||x1||=" << get_relative_norm(x1_2, x2_2) << '\n';
299.
300.
301. out << "---------------------------- A3 * x = b3 ---------------------------------\n";
302. out << "p3=" << p3 << "\n\n";
303.
304. out << "A3:\n";
305. show_matrix(A3);
306.
307. double* invA3 = inv(A3);
308. double* x1_3 = multiply_by_vector(invA3, b3);
309.
310. out << "x1(invA3*b3)=\n";
311. show_vector(x1_3);
312.
313. double* x2_3 = decomp_solve_2d(P3, b3);
314. out << "|x1-x2|=\n";
315. show_vector(minus(x1_3, x2_3));
316.
317.
318. out << "||x2-x1||/||x1||=" << get_relative_norm(x1_3, x2_3) << '\n';
319.
320.
321.
322. out << "---------------------------- A4 * x = b4 ---------------------------------\n";
323. out << "p4=" << p4 << "\n\n";
324.
325. out << "A4:\n";
```

```
show_matrix(A4);

double* invA4 = inv(A4);
double* x1_4 = multiply_by_vector(invA4, b4);

out << "x1(invA4*b4)=\n";
show_vector(x1_4);

double* x2_4 = decomp_solve_2d(P4, b4);
out << "|x1-x2|=\n";
show_vector(minus(x1_4, x2_4));


out << "||x2-x1||/||x1||=" << get_relative_norm(x1_4, x2_4) << '\n';


out << "---------------------------- A5 * x = b5 ---------------------------------\n";
out << "p5=" << p5 << "\n\n";

out << "A5:\n";
show_matrix(A5);

double* invA5 = inv(A5);
double* x1_5 = multiply_by_vector(invA5, b5);

out << "x1(invA5*b5)=\n";
show_vector(x1_5);

double* x2_5 = decomp_solve_2d(P5, b5);
out << "|x1-x2|=\n";
show_vector(minus(x1_5, x2_5));

out << "||x2-x1||/||x1||=" << get_relative_norm(x1_5, x2_5) << '\n';
}
```

Вывод

Исходя из таблицы 1, можно сказать, что при изменении параметра p в меньшую сторону увеличивается и число обусловленности(cond), и величины delta=||x1-x2||/||x1||.