

## Отчет по лабораторной работе №2

Дисциплина: Вычислительная математика

Вариант: 13

Выполнил

студент гр. 3530901/90003

\_\_\_\_\_

(подпись)

Руднев А.К.

Преподаватель

\_\_\_\_\_

(подпись)

Цыган В.Н.

«\_\_\_» \_\_\_\_\_ 2021 г.

### Условия задачи (вариант 18):

$$\begin{pmatrix} p-3 & -4 & -4 & 7 & 2 & 3 & 8 & 7 \\ 0 & -15 & -1 & 5 & -3 & 6 & 6 & -6 \\ -4 & 2 & -16 & 7 & 0 & 8 & -7 & 6 \\ 0 & 8 & -5 & -11 & 1 & 0 & 4 & 5 \\ 8 & 6 & -8 & 4 & 27 & -7 & -1 & 5 \\ -4 & -2 & 1 & 2 & -8 & 10 & 7 & 0 \\ 0 & -1 & 5 & 2 & -8 & 2 & -2 & 0 \\ 0 & -8 & -7 & 3 & -7 & -4 & -8 & 5 \end{pmatrix}$$

Рисунок 1 – Матрица, зависящая от  $p$

Для матриц, зависящих от параметра  $p$  ( $p = 1.0, 0.1, 0.01, 0.0001, 0.000001$ ), используя программы DECOMP и SOLVE, исследовать связь числа обусловленности ( $cond_c$ ) с нормой матрицы невязки  $R = E - A^{-1}A$ .

### Решение:

Задание будет выполняться на языке C++ при помощи программного обеспечения CLion.

- 1) Создам массив  $pValues$ , который включает в себя все 5 значений параметра  $p$ . Чтобы пройти по всем значениям параметра  $p$ , основное решение будет помещено в цикл:
- 2) Создам переменные, которые будут необходимы в работе, а также введу матрицу, заданную условием.
- 3) С помощью подпрограммы DECOMP получу LU – разложение матрицы  $A$ .
- 4) С помощью подпрограммы SOLVE будем  $N$  раз решаем уравнение  $Ax_k = e_k$ , где  $e_k$  и  $x_k$  -  $k$ -ые столбцы единичной и искомой обратной матрицы.
- 5) Транспонируем полученную матрицу и получаем обратную  $A^{-1}$
- 6) Вычисляем матрицу  $R = E - A^{-1}A$ : сначала вычисляем произведение  $M = A^{(-1)} * A$ , потом  $R = E - M$
- 7) Вычисляем норму матрицы

### 2. Листинг:

```

1  #include <iostream>
2  #include "MATRIX.H"
3  #include <windows.h>
4
5  #define n 8
6  #define N_TYPE double
7
8  int main() {
9      SetConsoleOutputCP(CP_UTF8);
10     //Массив из параметров
11     N_TYPE pArray[5] = {1, 0.1, 0.01, 0.0001, 0.000001};
12
13     //Цикл для разных параметров p
14     for (double p : pArray) {
15
16         //Параметр p
17         //Матрица A
18         double A[8][8] = {{p - 3, -4, -4, 7, 2, 3, 8, 7},
19                             {0, -15, -1, 5, -3, 6, 6, -6},
20                             {-4, 2, -16, 7, 0, 8, -7, 6},
21                             {0, 8, -5, -11, 1, 0, 4, 5},
22                             {8, 6, -8, 4, 27, -7, -1, 5},
23                             {-4, -2, 1, 2, -8, 10, 7, 0},
24                             {0, -1, 5, 2, -8, 2, -2, 0},
25                             {0, -8, -7, 3, -7, -4, -8, 5}};
26
27         //Переменные для работы с исходной системой
28         N_TYPE *x1; //Результат вызова Solve
29         N_TYPE cond = 0; //Обусловленность матрицы
30         N_TYPE work[n];
31         int ipvt[n];
32
33         //Переменные для работы с новой системой;
34         double A_LU[8][8]; //Для LU-разложения
35         double A_ob[8][8]; //Обратная матрица A-LU
36         double A_ob_tr[8][8]; //Транспонированная матрица A_ob
37         double M[8][8]; // M = A*A^(-1)
38         double R[8][8]; //Матрица невязки R = E - A*A^(-1)
39         double E[8][8]; //Единичная матрица
40
41         //Переменные для нахождения нормы матрицы
42         double temp;
43         double max = 0;
44
45
46         //Создание A_LU копии исходной матрицы A
47         for (int i = 0; i < n; i++) {
48             //LU-разложение матрицы
49             for (int i = 0; i < n; i++) {
50                 work[i] = 0;
51                 ipvt[i] = 0;
52             }
53             decomp(n, A_LU, &cond, ipvt, work);
54
55             //Вывод значения обусловленности в консоль
56             std::cout << "Параметр p = " << p << std::endl;
57             std::cout << "Обусловленность матрицы = " << cond << std::endl;
58
59             //Поиск элементов обратной матрицы
60             for (int i = 0; i < n; i++) {
61                 solve(n, A_LU, *A_ob, ipvt);
62             }
63
64             //Транспонируем матрицу
65             for (int i = 0; i < n; i++) {
66                 for (int j = 0; j < n; j++) {...}
67             }
68
69             // M = A*A^(-1)
70             for (int i = 0; i < n; i++) {
71                 for (int j = 0; j < n; j++)
72                     for (int k = 0; k < n; k++) {
73                         M[i][j] += A[i][k] * A_ob_tr[k][j];
74                     }
75             }
76         }
77     }
78 }

```

```

91
92 //R = E - M
93 for (int i = 0; i < n; i++) {
94     for (int j = 0; j < n; j++)
95         R[i][j] = E[i][j] - M[i][j];
96 }
97
98 //Вычисление нормы матрицы
99 for (auto & i : R){
100     temp = 0;
101     for (double j : i){
102         temp += fabs(j);
103     }
104     if (temp > max)
105         max = temp;
106 }
107
108 //Вывод нормы матрицы в консоль
109 std::cout << "Норма матрицы невязки R = " << max << std::endl;
110 }
111 }

```

### 3. Результаты работы программы:

На рисунке 1 представлена информация, которая выводится в консоль после запуска программы.

```

Параметр p = 1
Обусловленность матрицы = 1039.25
Норма матрицы невязки R = 3.06422e-14

Параметр p = 0.1
Обусловленность матрицы = 11830.2
Норма матрицы невязки R = 6.18172e-13

Параметр p = 0.01
Обусловленность матрицы = 120580
Норма матрицы невязки R = 5.96856e-12

Параметр p = 0.001
Обусловленность матрицы = 1.20818e+06
Норма матрицы невязки R = 4.41105e-11

Параметр p = 1e-06
Обусловленность матрицы = 1.20844e+09
Норма матрицы невязки R = 6.42613e-08

```

Рис. 1 – Результаты работы программы

#### 4. Установление связи между числом обусловленности $\text{cond}_A$ и нормой матрицы невязки $R = E - A^{-1}A$ :

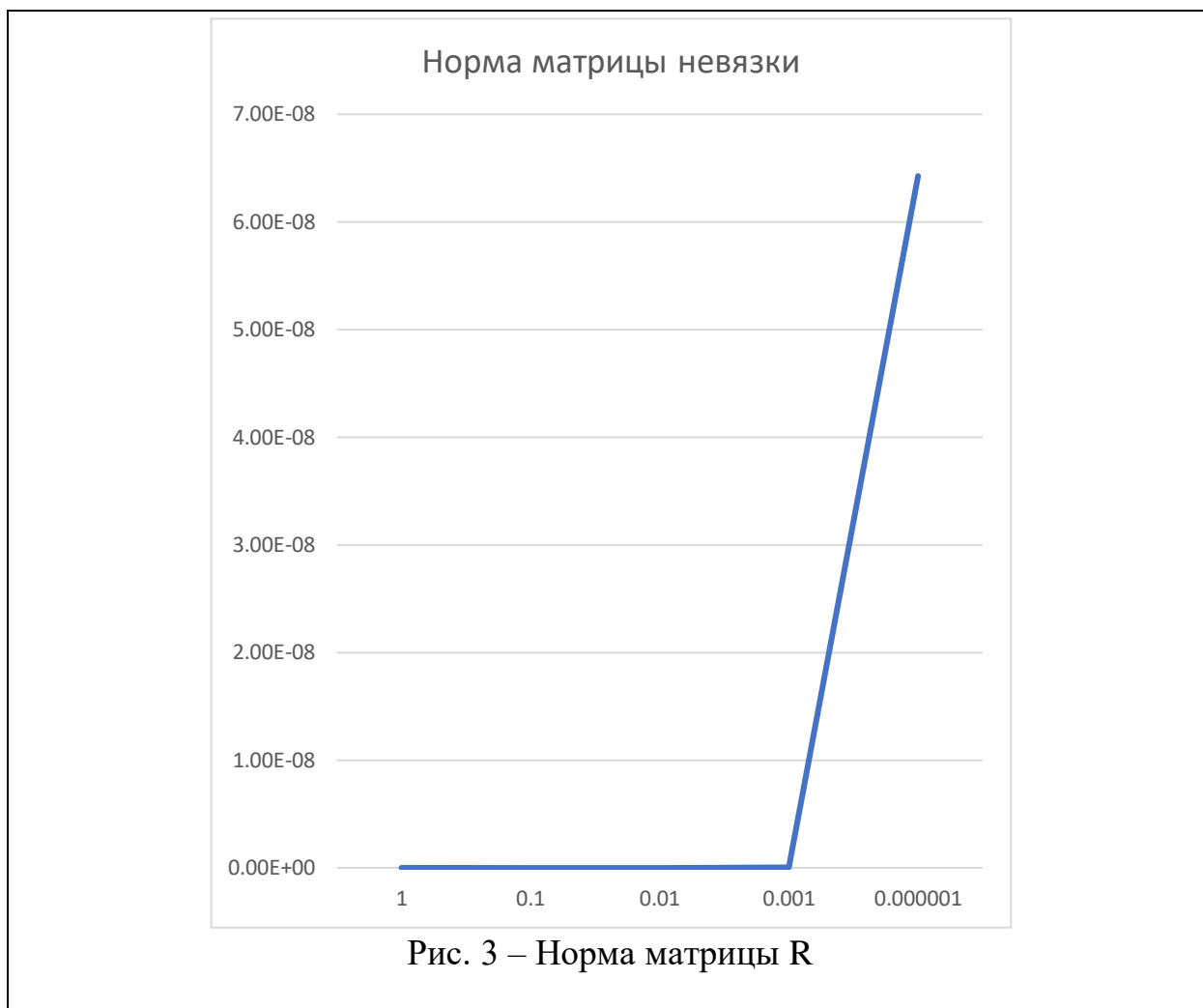
Составлю таблицу на основе полученных данных (таблица 1).

Составлю графики зависимости  $\text{cond}_A$  от параметра  $p$ , а также  $R$  от параметра  $p$ . (графики представлены на рисунках 2 и 3 соответственно).

Таблица 1 – Полученные данные

P	$\text{cond}_A$	R
1	1039.25	3.06422e-14
0.1	11830.2	6.18172e-13
0.01	120580	5.96856e-12
0.001	1.20842E+07	4.41105e-11
0.000001	1.20844E+09	6.42613e-08





Исходя из построенных графиков можно сделать вывод, что при уменьшении параметра  $p$  увеличивается значение обусловленности матрицы  $A$ , а также по приблизительно такому же графику увеличивается значение нормы матрицы невязки  $R$ . Для более подробных выводов построю дополнительные графики с новыми параметрами  $p$ , лежащим в интервале  $[0.000001, 0.000003]$  с промежутком в  $0.0000005$ .

Таблица 2 – Результаты вычислений

$P$	$\text{cond}_A$	$R$
0,0000010	1208440000	6.42613e-08
0,0000015	805630000	3.11993e-08
0,0000020	604222000	3.58559e-08
0,0000025	483378000	2.79397e-08
0,0000030	402815000	2.32831e-08

На рисунках 4 и 5 представлены новые график, исходя из результатов таблицы 2, полученных для  $p$ , лежащим в интервале  $[0.000001, 0.000003]$  с промежутком в  $0.0000005$ .

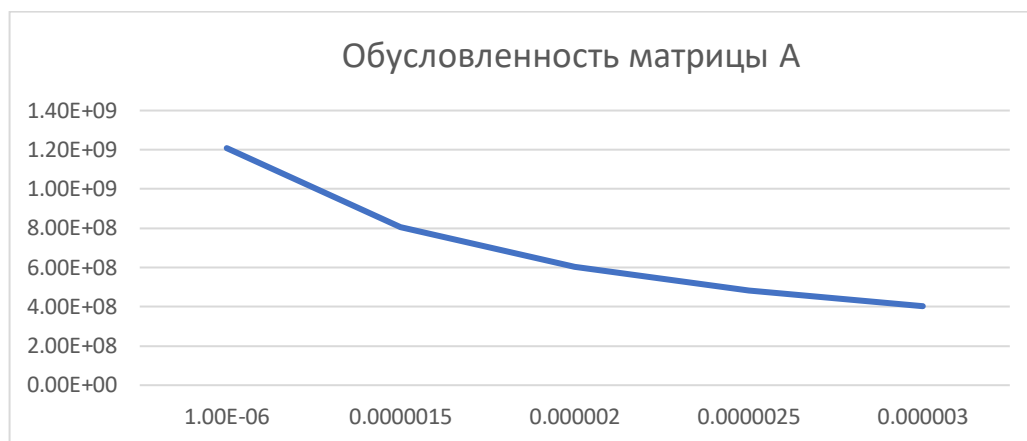


Рис. 4 – Обусловленность матрицы A



Рис. 5 – Норма матрицы невязки R

По рисункам 4 и 5 можно сказать, что график изменения обусловленности матрицы A похож на график нормы матрицы невязки – постепенно уменьшается при увеличении параметра.

**Вывод:** в ходе выполнения работы был проведен анализ связи обусловленности матрицы A и нормы матрицы невязки R. Исходя из полученных результатов можно сказать, что если происходит увлечение параметра  $p$ , то происходит и увлечение значения обусловленности матрицы A, а также значения нормы матрицы R, а при уменьшении параметра  $p$  происходит соответственное уменьшение обусловленности и нормы.