Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра Систем Штучного Інтелекту

Звіт

до лабораторної роботи № 1

з дисципліни

Чисельні методи

на тему:

“Метод Гауса для розв’язування систем лінійних алгебричних рівнянь”

Варіант №2

Виконав: студент КН-217

Ратушняк Денис

Прийняла: доцент каф. СШІ

Мочурад Л. І.

Львів – 2022

Мета роботи: засвоїти основні способи практичного використання методу Гауса.

Завдання 1.2. Скласти програму, яка знаходить обернену матрицю до заданої з використанням методу Гауса з постовпцевим вибором головного елемента. Представити детально документовану програму з результатами для наступних матриць:

$$\begin{pmatrix} 1 & 2 & -1 & -2 \\ 3 & 8 & 0 & -4 \\ 2 & 2 & -4 & -3 \\ 3 & 8 & -1 & -6 \end{pmatrix}, \qquad \begin{pmatrix} 0 & 1 & 3 \\ 2 & 2 & 5 \\ 3 & 5 & 7 \end{pmatrix}, \qquad \begin{pmatrix} 0 & 0 & 1 & -1 \\ 0 & 3 & 1 & 4 \\ 2 & 7 & 6 & -1 \\ 1 & 2 & 2 & -1 \end{pmatrix}.$$

Програмна реалізація на мові програмування C++

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long double ld;
typedef long long ll;
typedef vector< vector<ld > > matrix;
const ld eps = 1e-10;
ifstream tests("input.txt");
ofstream answers("output.txt");
void print(matrix &A)
{
    ll n = A.size();
    for(int i = 0; i < n; ++i){
        for(int j = 0; j < A[i].size(); ++j) answers << A[i][j] << " ";
        answers << "\n";
    }
    answers << "\n";
}

matrix operator * (const matrix &A, const matrix &B)
{
    ll n = A.size();
    matrix res(n, vector<ld>(n, 0.0));
    for(int k = 0; k < n; ++k)
        for(int i = 0; i < n; ++i)
            for(int j = 0; j < n; ++j)
                res[i][j] += A[i][k] * B[k][j];
    return res;
}

pair<bool, matrix> get_inverse_matrix(matrix A)
{
    ll n = A.size();
    matrix AE;
    AE.resize(n);
```

```cpp
    /// forming matrix A|E
    for(int i = 0; i < n; ++i){
        for(int j = 0; j < n; ++ j) AE[i].emplace_back(A[i][j]);

        for(int j = 0; j < n; ++ j) AE[i].emplace_back(i == j);
    }
    for(int k = 0; k < n; ++k)
    {
        int row_max_el = k;

        /// finding maximum element
        for(int i = k + 1; i < n; ++i)
            if(fabs(AE[i][k]) > fabs(AE[row_max_el][k])) row_max_el = i;

        /// if maximum element goes to 0 -> inverse matrix does not exist
        if(fabs(AE[row_max_el][k]) < eps) return {false, A};


        /// change of 2 rows for optimization of division by the maximum element
        if(k != row_max_el)
            for(int j = k; j < 2 * n; ++j) swap(AE[k][j], AE[row_max_el][j]);

        /// forming zeros under main diagonal
        for(int i = k + 1; i < n; ++i){
            ld mik = -AE[i][k]/AE[k][k];

            AE[i][k] = 0;
            for(int j = k + 1; j < 2 * n; ++j) AE[i][j] += mik * AE[k][j];
        }
    }

    for(int k = n - 1; k >= 0; --k){

        /// making 1 on the main diagonal
```

```cpp
        for(int j = n; j < 2 * n; ++ j) AE[k][j] /= AE[k][k];
        AE[k][k] = 1;

        /// making 0 above main diagonal
        for(int i = k - 1; i >= 0; --i){
            ld mn = -AE[i][k];
            AE[i][k] = 0;
            for(int j = n; j < 2 * n; ++j)
            {
                AE[i][j] += AE[k][j] * mn;
            }
        }
    }

    ///forming inverse matrix
    for(int i = 0; i < n; ++i)
        for(int j = 0; j < n; ++j)
            A[i][j] = AE[i][j + n];

    return {true, A};
}

matrix get_random_matrix(ll size_)
{
    ///randomize due to clock
    srand(clock());
    matrix res;
    res.resize(size_);
    for(int i = 0; i < size_; ++i){
        res[i].resize(size_);
        for(int j = 0; j < size_; ++j) res[i][j] = rand() - rand()/2;
    }
    return res;
}
```

```cpp

void solve(matrix &A)
{
    ll n = A.size();
    pair<bool, matrix> inv_A = get_inverse_matrix(A);
    answers << "Our matrix:\n";
    print(A);
    if(inv_A.first)
    {
        answers << "Inverse matrix:\n";
        print(inv_A.second);

        answers << "The product of the given matrix and its inverse:\n";
        matrix E = A * inv_A.second;
        print(E);

        ld sum = 0.0;
        /// Calculating deviation - Frobenius norm
        for(int i = 0; i < n; ++i)
            for(int j = 0; j < n; ++j)
                sum += (E[i][j] - (i == j)) * (E[i][j] - (i == j));

        answers << "Deviation from the unit matrix = Frobenius norm = " << sqrt(sum) << "\n";

        ld norm_A = 0, norm_invA = 0;
        ld sum_A, sum_invA;
        /// Calculating norms and cond
        for(int i = 0; i < n; ++i)
        {
            sum_A = 0;
            sum_invA = 0;

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

```cpp
                sum_A += fabs(A[i][j]);
                sum_invA += fabs(inv_A.second[i][j]);
            }

            norm_A = max(norm_A, sum_A);
            norm_invA = max(norm_invA, sum_invA);
        }

        answers << "Norm of the given matrix = " << norm_A << "\n";
        answers << "Norm of the inverse matrix = " << norm_invA << "\n";
        answers << "Cond of the given matrix = " << norm_A * norm_invA << "\n";
    }
    else answers << "The inverse matrix cannot be found, since the determinant of this matrix is zero\n";
    answers << "-------------------------------------------------------\n\n";
}

int main()
{
    ll n;
    while(tests >> n){
        matrix A(n, vector<ld>(n, 0));
        for(int i = 0; i < n; ++i)
            for(int j = 0; j < n; ++j)
                tests >> A[i][j];
        solve(A);
    }
    ll cnt_of_random_matrix = 5;
    while(cnt_of_random_matrix--){
        matrix A = get_random_matrix(cnt_of_random_matrix  + rand() % 5);
        solve(A);
    }
    return 0;
}
```

Вхідні дані

```
4
1 2 -1 -2
3 8 0 -4
2 2 -4 -3
3 8 -1 -6

3
0 1 3
2 2 5
3 5 7

4
0 0 1 -1
0 3 1 4
2 7 6 -1
1 2 2 -1

5
1 2 3 4 5
3 6 9 12 15
6 2 0 2 -1
2 1 2 0 2
3 4 1 -2 1
```

Вихідні дані

```
Our matrix:
1 2 -1 -2
3 8 0 -4
2 2 -4 -3
3 8 -1 -6

Inverse matrix:
24 3 -4 -8
-11.5 -1 2 3.5
10 1 -2 -3
-5 2.71051e-19 1 1

The product of the given matrix and its inverse:
1 -2.1684e-19 0 3.25261e-19
-3.46945e-18 1 -2.1684e-19 -2.1684e-19
-2.60209e-18 -3.79471e-19 1 6.50521e-19
-3.46945e-18 -2.1684e-19 -4.33681e-19 1

Deviation from the unit matrix = Frobenius norm = 5.93174e-18
Norm of the given matrix = 18
Norm of the inverse matrix = 39
Cond of the given matrix = 702
-------------------------------------------------------

Our matrix:
0 1 3
2 2 5
3 5 7

Inverse matrix:
-0.846154 0.615385 -0.0769231
0.0769231 -0.692308 0.461538
0.307692 0.230769 -0.153846

The product of the given matrix and its inverse:
1 0 0
1.0842e-19 1 0
0 0 1

Deviation from the unit matrix = Frobenius norm = 1.53329e-19
Norm of the given matrix = 15
Norm of the inverse matrix = 1.53846
Cond of the given matrix = 23.0769
-------------------------------------------------------
```

```
Our matrix:
0 0 1 -1
0 3 1 4
2 7 6 -1
1 2 2 -1

Inverse matrix:
-0.166667 0.5 -1.16667 3.33333
-1.16667 -0.5 0.833333 -1.66667
1.5 0.5 -0.5 1
0.5 0.5 -0.5 1

The product of the given matrix and its inverse:
1 0 0 0
0 1 0 0
0 0 1 0
-2.1684e-19 0 1.0842e-19 1

Deviation from the unit matrix = Frobenius norm = 3.25261e-19
Norm of the given matrix = 16
Norm of the inverse matrix = 5.16667
Cond of the given matrix = 82.6667
-------------------------------------------------------

Our matrix:
1 2 3 4 5
3 6 9 12 15
6 2 0 2 -1
2 1 2 0 2
3 4 1 -2 1

The inverse matrix cannot be found, since the determinant of this matrix is zero
-------------------------------------------------------
```

```
Our matrix:
-3821 20020 2957 -7777 -4856
-8197 1002 12577 -4197 -4343
5031 8702 17796 27761 -6414
2151 -8453 6289 17303 12023
26094 19280 21305 -3343 -955

Inverse matrix:
-3.29355e-05 -2.87823e-05 2.26371e-06 -2.04496e-05 2.5708e-05
5.75683e-05 -2.36847e-05 2.49866e-06 1.59647e-05 -8.08061e-07
-9.08118e-06 5.28321e-05 -2.35658e-06 1.53078e-05 1.44591e-05
4.16884e-06 -2.27165e-05 2.77192e-05 7.27509e-06 -1.24696e-05
4.51175e-05 -6.44539e-06 -3.73079e-05 7.95796e-05 5.21504e-06

The product of the given matrix and its inverse:
1 2.71051e-20 1.35525e-20 -5.42101e-20 -1.35525e-20
1.35525e-20 1 -1.35525e-20 -2.71051e-20 -1.01644e-20
-2.71051e-20 2.71051e-20 1 0 3.38813e-20
-5.42101e-20 -2.03288e-20 0 1 0
7.45389e-20 1.2282e-20 0 -6.77626e-21 1

Deviation from the unit matrix = Frobenius norm = 2.01192e-19
Norm of the given matrix = 70977
Norm of the inverse matrix = 0.000173665
Cond of the given matrix = 12.3263
-------------------------------------------------------
```

```
Our matrix:
-9192 -6916 11307 -3201 14730 -8367 14868
9716 -2976 2109 25089 3826 -5899 8063
9771 -5416 9002 -8087 12518 12088 16798
16156 15341 11286 24819 11725 16284 13608
3853 27140 10799 21269 -14765 4569 7940
15544 12526 16695 -544 22619 13318 14107
12836 1278 777 725 18965 -5232 15077

Inverse matrix:
-4.92245e-05 5.74542e-05 2.04695e-05 -6.58654e-05 1.08207e-05 5.25903e-05 -4.47134e-07
1.745e-06 -4.21127e-05 -3.0733e-05 1.74199e-05 1.66881e-05 -3.43154e-06 3.37413e-05
4.91298e-06 6.9882e-05 4.50226e-06 -7.69281e-05 1.45327e-05 9.12142e-05 -7.07996e-05
1.0294e-05 1.17786e-05 -1.47524e-05 3.33913e-05 -9.98244e-06 -1.61664e-05 -9.76857e-06
1.8375e-05 -2.40425e-05 -3.96625e-05 4.02552e-05 -3.13307e-05 4.33153e-06 1.50411e-05
-6.86419e-08 -3.89921e-05 2.08238e-05 6.08551e-05 -1.75073e-05 -3.53776e-05 -1.4885e-05
1.78746e-05 -3.2801e-05 4.27723e-05 2.74395e-05 2.24389e-05 -6.61312e-05 4.388e-05

The product of the given matrix and its inverse:
1 1.6263e-19 5.42101e-20 2.71051e-20 8.13152e-20 5.42101e-20 -2.1684e-19
1.35525e-20 1 2.71051e-20 -1.21973e-19 0 0 2.71051e-20
-2.71051e-20 2.1684e-19 1 -2.71051e-20 2.71051e-20 1.0842e-19 0
-2.71051e-20 5.42101e-20 0 1 0 -5.42101e-20 0
-4.06576e-20 -5.42101e-20 -2.71051e-20 -5.42101e-20 1 -5.42101e-20 2.71051e-20
-8.13152e-20 8.13152e-20 -5.42101e-20 -1.0842e-19 8.13152e-20 1 -5.42101e-20
0 1.0842e-19 -5.42101e-20 -1.6263e-19 -2.71051e-20 0 1

Deviation from the unit matrix = Frobenius norm = 5.58621e-19
Norm of the given matrix = 109219
Norm of the inverse matrix = 0.000332772
Cond of the given matrix = 36.345
-------------------------------------------------------
```

```
Our matrix:
-9192 -6916 11307 -3201 14730 -8367
14868 9716 -2976 2109 25089 3826
-5899 8063 9771 -5416 9002 -8087
12518 12088 16798 16156 15341 11286
24819 11725 16284 13608 3853 27140
10799 21269 -14765 4569 7940 15544

Inverse matrix:
-7.3858e-05 6.04075e-05 1.57122e-05 -9.82889e-06 1.06029e-05 -5.78267e-05
-3.72668e-05 -1.03088e-05 5.34033e-05 1.55178e-05 -9.66681e-06 1.58727e-05
2.20467e-06 -9.35316e-06 3.22738e-05 -1.71961e-06 2.32656e-05 -1.90937e-05
-2.26881e-05 -1.52867e-05 -4.78802e-05 9.04172e-05 -5.47221e-05 -3.46392e-06
4.71013e-05 1.42568e-05 -1.75169e-05 -1.61647e-06 -2.05717e-06 1.74965e-05
8.70079e-05 -3.95352e-05 -3.031e-05 -4.17896e-05 4.50964e-05 5.67331e-05

The product of the given matrix and its inverse:
1 5.42101e-20 -2.71051e-20 -5.42101e-20 0 2.71051e-20
1.6263e-19 1 7.45389e-20 -2.71051e-20 -2.71051e-20 -9.48677e-20
5.42101e-20 5.42101e-20 1 0 5.42101e-20 5.42101e-20
-5.42101e-20 -8.13152e-20 2.71051e-20 1 0 -5.42101e-20
2.1684e-19 -1.0842e-19 5.42101e-20 0 1 1.0842e-19
1.0842e-19 -1.6263e-19 8.13152e-20 -5.42101e-20 0 1

Deviation from the unit matrix = Frobenius norm = 4.65397e-19
Norm of the given matrix = 97429
Norm of the inverse matrix = 0.000300472
Cond of the given matrix = 29.2747
-------------------------------------------------------

Our matrix:
-9192

Inverse matrix:
-0.00010879

The product of the given matrix and its inverse:
1

Deviation from the unit matrix = Frobenius norm = 0
Norm of the given matrix = 9192
Norm of the inverse matrix = 0.00010879
Cond of the given matrix = 1
-------------------------------------------------------
```

```
Our matrix:
-9192 -6916 11307 -3201
14730 -8367 14868 9716
-2976 2109 25089 3826
-5899 8063 9771 -5416

Inverse matrix:
-1.15497e-05 8.33278e-05 -8.23858e-05 9.81121e-05
-7.27774e-05 -2.85855e-05 4.15324e-05 2.10721e-05
1.51765e-05 2.55798e-05 2.73023e-06 3.88476e-05
-6.83868e-05 -8.71669e-05 0.00015649 -0.000190044

The product of the given matrix and its inverse:
1 -1.0842e-19 0 -1.0842e-19
0 1 0 0
-2.71051e-20 -5.42101e-20 1 5.42101e-20
-2.71051e-20 -2.71051e-20 0 1

Deviation from the unit matrix = Frobenius norm = 1.85823e-19
Norm of the given matrix = 47681
Norm of the inverse matrix = 0.000502087
Cond of the given matrix = 23.94
-------------------------------------------------------
```

Аналіз чисельних експериментів

З отриманих результатів можна зробити наступні висновки:

1) Числа зумовленості кожної з заданих матриць є великими(Оскільки вони є значно більшими за 1).

2) Перевірка на точність була наступною. Спочатку була порахована матриця $\tilde{E}$ як добуток вхідної матриці на обернену до неї. Далі було пораховане відхилення матриці $\tilde{E}$ - E. Де E – одинична матриця. Відхилення є досить малим відносно чисел в початковій матриці, отже усі обрахунки були виконані досить точно.

3) Чим більше розмір матриці і чим більші числа задані, тим більшим буде число зумовленості і похибка обчислень.

Висновок

Я закріпив вміння та навички практичного використання методу Гауса. Написав програму, яка по заданим розмірам матриць і самим матрицям знаходить обернену до них, або виводить, що знайти таку матрицю неможливо. Також у випадку існування оберненої матриці програма виводить добуток матриці на її обернену, норми цих матриць, число зумовленості матриці та відхилення добутку матриць від одиничної матриці(норму Фробеніуса). Також в кінці програма пророблює всі операції над 5-ма випадково згенерованими матрицями.