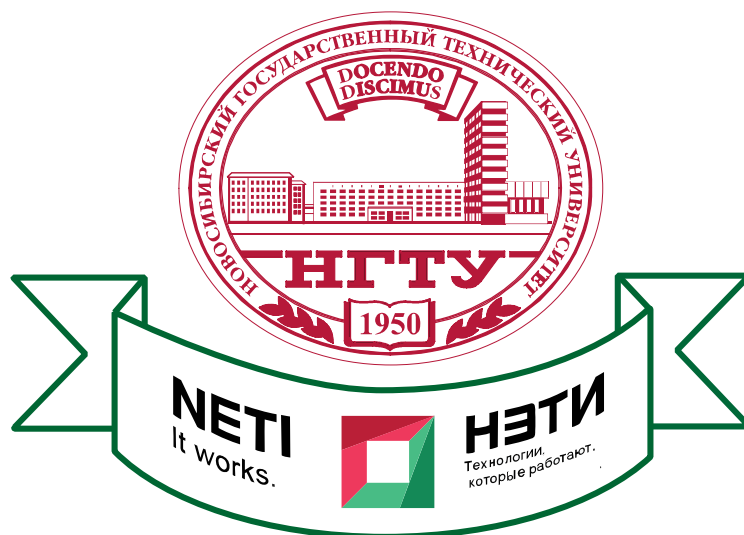


Министерство науки и высшего образования
Российской Федерации

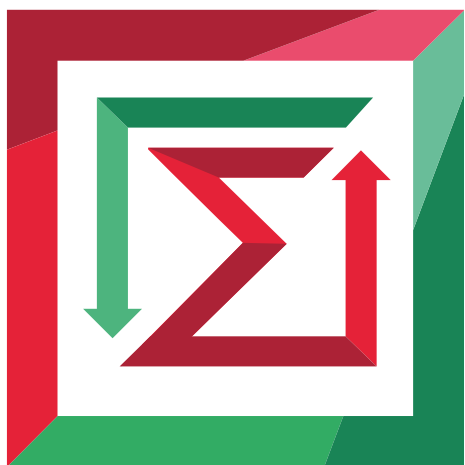
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



Кафедра прикладной математики

Практическое задание № 5
по дисциплине «Численные методы»

СОБСТВЕННЫЕ ЧИСЛА



Факультет:	ПМИ
Группа:	ПМ-71
Студент:	Востриков Вячеслав
Преподаватели:	Патрушев И.И. Задорожный А.Г. Персова М.Г.

Новосибирск
2020

1. Цель

Реализовать степенной метод для поиска максимума и модифицировать его для поиска минимума (без обращения матрицы, т.е. через LU-разложение).

2. Анализ

Для нахождения максимального собственного значения будем использовать степенной метод. Для того нахождения минимального собственного значения будем находить максимальное собственное значение обратной матрицы.

Поиск модуля максимального значения:

Генерируем случайный ненулевой вектор x^0 . В данном случае берём вектор $x^0 = [0, 1, \dots, n - 1]$.

Для $k = 1, 2, \dots$ вычисляются:

$$x^k = Ax^{k-1}$$
$$|\lambda_k| = \frac{\|x^k\|}{\|x^{k-1}\|}$$

Чтобы не было переполнения, каждые 5 итераций цикла

$$x^k = \frac{x_i^k}{\|x^k\|}, i = \overline{1, n}$$

если $\left| \frac{\lambda_k - \lambda_{k-1}}{\lambda_k} \right| < \text{eps}$ и $k < \text{max_iter}$, то продолжаем итерации

Результат итерационного процесса:

$$|\lambda_{\max}| \cong |\lambda_k|$$

Поиск модуля минимального значения:

Строим обратную матрицу и делаем поиск по модулю максимального значения.

3. Текст программы

```
#include <string>
#include <fstream>
#include <locale>
#include <iostream>
#include <iomanip>

using namespace std;

#define N 10

typedef double T;
```

```

int n, max_iter = 10000;
T eps = 1e-14, max, min;
T* x1, *x;
T** A;

void Hylbert_m(const int size)
{
    A = new T* [size];
    for (int i = 0; i < size; i++)
        A[i] = new T[size];

    for (int i = 0; i < size; i++)
        for (int j = 0; j < size; j++)
            A[i][j] = 1. / T(i + j + 1);

    ofstream fout("hylb.txt");
    for (int i = 0; i < size; i++)
    {
        for (int j = 0; j < size; j++)
            fout << setprecision(15) << A[i][j] << " ";
        fout << endl;
    }
}

void input()
{
    ifstream fcin("matrix_info.txt");
    A = new T* [n];
    x = new T[n];
    x1 = new T[n];
    fcin >> n;
    for (int i = 0; i < n; i++)
    {
        A[i] = new T[n];
        for (int j = 0; j < n; j++)
            fcin >> A[i][j];
    }
    fcin.close();
}

void output()
{
    ofstream fout("result.txt");
    fout << setprecision(15) << "MAX_VALUE: " << to_string(max) << endl << "MIN_VALUE: " << to_string(min);
    fout.close();
}

T norm_vec(T* vec)
{
    T sum = 0;
    for (int i = 0; i < n; i++)
        sum += vec[i] * vec[i];
    return sqrt(sum);
}

void LU_dec()
{

```

```

for (int i = 0; i < n; i++)
{
    for (int j = i; j < n; j++)
    {
        T sum = 0;
        for (int k = 0; k < i; k++)
            sum += A[j][k] * A[k][i];
        A[j][i] -= sum;
    }

    for (int j = i + 1; j < n; j++)
    {
        T sum = 0;
        for (int k = 0; k < i; k++)
            sum += A[i][k] * A[k][j];
        A[i][j] = (A[i][j] - sum) / A[i][i];
    }
}

void solve_LU(T* res)
{
    for (int i = 0; i < n; i++)
    {
        T sum = 0;
        for (int j = 0; j < i; j++)
            sum += A[i][j] * res[j];
        res[i] = (x[i] - sum) / A[i][i];
    }
    for (int i = n - 1; i >= 0; i--)
    {
        T sum = 0;
        for (int j = i + 1; j < n; j++)
            sum += A[i][j] * res[j];
        res[i] -= sum;
    }
}

void MAX_eigenvalue()
{
    int iter;
    T norm_1, norm_2, supp_val;
    T lambda_1, lambda = 0;
    bool end = false;

    for (int i = 0; i < n; i++)
        x[i] = i;

    norm_1 = norm_vec(x);

    for (iter = 0; iter < max_iter && !end; iter++)
    {
        for (int i = 0; i < n; i++)
        {
            x1[i] = 0;

            for (int j = 0; j < n; j++)
                x1[i] += A[i][j] * x[j];
        }
        norm_2 = norm_vec(x1);
        lambda_1 = norm_2 / norm_1;
    }
}

```

```

    supp_val = fabs((lambda_1 - lambda) / lambda_1);

    if (supp_val < eps)
        end = true;

    for (int i = 0; i < n; i++)
        x[i] = x1[i];
    lambda = lambda_1;
    norm_1 = norm_2;

    if (iter % 5 == 0)
    {
        for (int i = 0; i < n; i++)
            x[i] /= norm_1;
        norm_1 = 1;
    }
}

cout << "iter_max = " << to_string(iter) << endl;
max = lambda;
}

```

```

void MIN_eigenvalue()
{
    int iter;
    T norm_1, norm_2, supp_val;
    T lambda_1, lambda = 0;
    bool end = false;

    for (int i = 0; i < n; i++)
        x[i] = i;

    norm_1 = norm_vec(x);

    LU_dec();

    for (iter = 0; iter < max_iter && !end; iter++)
    {
        solve_LU(x1);
        norm_2 = norm_vec(x1);
        lambda_1 = norm_2 / norm_1;

        supp_val = fabs((lambda_1 - lambda) / lambda_1);

        if (supp_val < eps)
            end = true;

        x = x1;
        lambda = lambda_1;
        norm_1 = norm_2;

        if (iter % 5 == 0)
        {
            for (int i = 0; i < n; i++)
                x[i] /= norm_1;
            norm_1 = 1;
        }
    }
    cout << "iter_min = " << to_string(iter);
    min = 1. / lambda;
}

```

```

int main()
{
    int choice;
    setlocale(LC_ALL, "rus");
    cout << "Сгенерировать матрицу Гильберта? n - размерность матрицы, * <= 0 -
нет.\n";
    cin >> choice;
    if (choice > 0)
        Hylbert_m(choice);
    input();
    MAX_eigenvalue();
    MIN_eigenvalue();
    output();

    return 1;
}

```

4. Исследования

Во всех тестах $\text{eps} = 1\text{e-}14$, $\text{max_iter} = 10000$.

Матрица Гильберта ($n = 5, n = 10$)

Результаты Маткада:

$$\begin{aligned}
 \underline{H1} &:= \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ 1 & 2 & 3 & 4 & 5 \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ 2 & 3 & 4 & 5 & 6 \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{pmatrix} & \begin{aligned} \max(\text{eigenvals}(H1)) &= 1.567 \\ \min(\text{eigenvals}(H1)) &= 3.288 \times 10^{-6} \end{aligned} \\
 \underline{H2} &:= \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} \\ 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} \\ \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} & \frac{1}{16} \\ \frac{1}{7} & \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} & \frac{1}{16} & \frac{1}{17} \\ \frac{1}{8} & \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} & \frac{1}{16} & \frac{1}{17} & \frac{1}{18} \\ \frac{1}{9} & \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} & \frac{1}{16} & \frac{1}{17} & \frac{1}{18} & \frac{1}{19} \\ \frac{1}{10} & \frac{1}{11} & \frac{1}{12} & \frac{1}{13} & \frac{1}{14} & \frac{1}{15} & \frac{1}{16} & \frac{1}{17} & \frac{1}{18} & \frac{1}{19} & \frac{1}{20} \end{pmatrix} & \begin{aligned} \max(\text{eigenvals}(H2)) &= 1.752 \\ \min(\text{eigenvals}(H2)) &= 1.094 \times 10^{-13} \end{aligned}
 \end{aligned}$$

Результаты программы:

Для матрицы размерности $k = 5$

```
iter_max = 11 MAX_VALUE: 1,567051
iter_min = 6 MIN_VALUE: 0,000003
```

Для матрицы размерности $k = 10$

```
iter_max = 12 MAX_VALUE: 1,751920
iter_min = 6 MIN_VALUE: 0,000000
```

Матрица A^k ($k = 0, k = 10$)

Результаты Маткада:

$$A0 := \begin{pmatrix} 15 & -3 & -4 & -3 & -4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & -2 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ -3 & -4 & 15 & -1 & -3 & -1 & -3 & 0 & 0 & 0 \\ -4 & -4 & -1 & 17 & -1 & -1 & -4 & -2 & 0 & 0 \\ -2 & -2 & -1 & 0 & 15 & -2 & -2 & -3 & -3 & 0 \\ 0 & 0 & -2 & -1 & -4 & 13 & -2 & -2 & -1 & -1 \\ 0 & 0 & -2 & -3 & -4 & -4 & 17 & 0 & -2 & -2 \\ 0 & 0 & 0 & -3 & -1 & -4 & 0 & 15 & -4 & -3 \\ 0 & 0 & 0 & 0 & -4 & -2 & -2 & -3 & 11 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 & -2 & 0 & -4 & 8 \end{pmatrix}$$

$\max(\text{eigenvals}(A0)) = 23.569 + 0.371i$

$\min(\text{eigenvals}(A0)) = 0.05 - 0.371i$

$$A10 := \begin{pmatrix} 15.00000000001 & -3 & -4 & -3 & -4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & -2 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ -3 & -4 & 15 & -1 & -3 & -1 & -3 & 0 & 0 & 0 \\ -4 & -4 & -1 & 17 & -1 & -1 & -4 & -2 & 0 & 0 \\ -2 & -2 & -1 & 0 & 15 & -2 & -2 & -3 & -3 & 0 \\ 0 & 0 & -2 & -1 & -4 & 13 & -2 & -2 & -1 & -1 \\ 0 & 0 & -2 & -3 & -4 & -4 & 17 & 0 & -2 & -2 \\ 0 & 0 & 0 & -3 & -1 & -4 & 0 & 15 & -4 & -3 \\ 0 & 0 & 0 & 0 & -4 & -2 & -2 & -3 & 11 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 & -2 & 0 & -4 & 8 \end{pmatrix} +$$

$\max(\text{eigenvals}(A10)) = 23.569 + 0.371i$

$\min(\text{eigenvals}(A10)) = 0.05 - 0.371i$

Результаты программы:

Для матрицы $k = 0$

```
iter_max = 111 MAX_VALUE: 23,569117
iter_min = 10 MIN_VALUE: 0,049720
```

Для матрицы $k = 10$

```
iter_max = 112 MAX_VALUE: 23,569117
iter_min = 10 MIN_VALUE: 0,049720
```

Матрица с диагональным преобладанием

Результаты Маткада:

$$B := \begin{pmatrix} 9 & -4 & 0 & 0 & 0 & 0 & -1 & 0 & -3 & 0 & 0 & 0 \\ -2 & 8 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & -4 & 0 & 0 \\ 0 & -1 & 8 & -3 & 0 & 0 & 0 & 0 & 0 & 0 & -4 & 0 \\ 0 & 0 & 0 & 7 & -3 & 0 & 0 & 0 & 0 & -2 & 0 & -2 \\ 0 & 0 & 0 & -2 & 9 & -4 & 0 & 0 & 0 & 0 & -3 & 0 \\ 0 & 0 & 0 & 0 & 1 & 3 & -1 & 0 & 0 & 0 & 0 & -1 \\ -4 & 0 & 0 & 0 & 0 & 0 & 5 & -1 & 0 & 0 & 0 & 0 \\ 0 & -3 & 0 & 0 & 0 & 0 & -1 & 6 & -2 & 0 & 0 & 0 \\ -2 & 0 & -4 & 0 & 0 & 0 & 0 & 0 & 8 & -2 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & -4 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 0 & -3 & 8 & -1 \\ 0 & 0 & 0 & -4 & 0 & -2 & 0 & 0 & 0 & 0 & -3 & 9 \end{pmatrix}$$

$$\max(\text{eigenvals}(B)) = 14.802 + 2.325i$$

$$\min(\text{eigenvals}(B)) = 0.3 - 2.325i$$

Результаты программы:

```
iter_max = 247 MAX_VALUE: 14,802439
iter_min = 15 MIN_VALUE: 0,299751
```

Матрица с положительным знаком вне диагональных элементов

Результаты Маткада:

$$B1 := \begin{pmatrix} 9 & 4 & 0 & 0 & 0 & 0 & 1 & 0 & 3 & 0 & 0 & 0 \\ 2 & 8 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 4 & 0 & 0 \\ 0 & 1 & 8 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 7 & 3 & 0 & 0 & 0 & 0 & 2 & 0 & 2 \\ 0 & 0 & 0 & 2 & 9 & 4 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 1 & 3 & 1 & 0 & 0 & 0 & 0 & 1 \\ 4 & 0 & 0 & 0 & 0 & 0 & 5 & 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 1 & 6 & 2 & 0 & 0 & 0 \\ 2 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 8 & 2 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 4 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 3 & 8 & 1 \\ 0 & 0 & 0 & 4 & 0 & 2 & 0 & 0 & 0 & 0 & 3 & 9 \end{pmatrix}$$

$$\max(\text{eigenvals}(B1)) = 15.001 + 2.836i$$

$$\min(\text{eigenvals}(B1)) = 0.608 - 2.836i$$

+

Результаты работы программы:

```
iter_max = 199 MAX_VALUE: 15,000922  
iter_min = 81  MIN_VALUE: 0,608312
```