|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | |
| МИНОБРНАУКИ РОССИИ | | | |
| 1. Федеральное государственное бюджетное образовательное учреждение 2. высшего профессионального образования 3. **"Московский технологический университет"** 4. МИРЭА | | | |
| Институт Кибернетики | | |  |
| Кафедра программного обеспечения систем радиоэлектронной аппаратуры при АО «Концерн радиостроения «ВЕГА» | | |  |

|  |  |
| --- | --- |
| **Отчет по лабораторной работе №2** | |
| **по дисциплине** | |
| **«** Численные методы **»** | |
| **Вариант 5** | |
| Студент 3-го курса  группы КМБО-02-16 | Карабанов В.В. |
| Преподаватель | Даева С.Г. |
| Рецензент |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| Работа представлена к защите | «\_\_»\_\_\_\_\_\_\_2018 г. |  |
|  |  |  |
| «Допущен к защите» | «\_\_»\_\_\_\_\_\_\_2018 г. |  |

Москва 2018

Содержание

Задание №1 3

Теоретическая часть 3

Практическая часть 4

Задание №2 5

Теоретическая часть 5

Практическая часть 6

Задание №3 7

Теоретическая часть 8

Практическая часть 10

Приложения 12

# Задание №1

Разработать программу решения системы линейных уравнений (СЛУ) точным методом Халецкого.

## Теоретическая часть

Система линейных уравнений – это такая система уравнений, в которую входят неизвестные не более 1 степени.

, где

В точном методе решения алгоритм всегда имеет строго определенное количество шагов. В результате применения точного метода решение может состоять как из точных, так и из приближенных значений.

Решение СЛУ по схеме Халецкого.

## Практическая часть

## Задание 1

Для расчета последовательности была разработана функция “func\_lab2”. Для задач лабораторной работы была написана программа “lab2.1”. Программа написаны на языке «C++» в операционной системе “Windows” с использованием текстового редактора “Visual Studio 2017”. Для отладки проводились тесты программы на примерах с заранее известным ответом.

# Задание №2

Для решения СЛУ *АХ = В,* где А — квадратная невырожденная матрица размерности применяют метод Зейделя. Для этого необходимо преобразовать исходную систему. Каждую строку делят на ее диагональный элемент, затем все элементы, кроме диагональных, переносятся через знак равенства. Система принимает следующий вид где— столбец с элементами— матрица с элементамиДалее производят расчет последовательных приближенийдо тех пор, пока не будет достигнута необходимая точность.Рассчитываются последующим формулам:

Метод сходится независимо от выбора начального приближения, если есть какая-нибудь каноническая норма матрицыимеет значение меньше единицы, или если модули диагональных элементов матрицы *А* для каждой строки больше суммы остальных элементов строки.

## Практическая часть

Для вычисления определителей и выполнения других необходимых действий с матрицами был разработан класс “Matrix”. Определитель считается методом Гаусса.

В программе “lab2.2” последовательные приближениявычисляются пока где — наперед заданная точность, а норма вектора вычисляется как корень из суммы квадратов координат.

# Задание №3

Исследовать сходимость итерационных процессов для конкретной СЛУ и решить ее точным и приближенным методом, разработанным в заданиях 1 и 2. Сопоставить время расчетов и точность результатов.

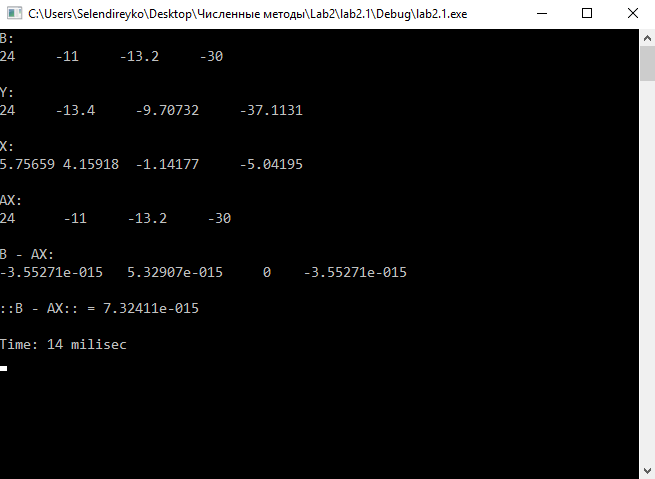
## Практическая часть

Дана следующая матрица и столбец свободных членов:

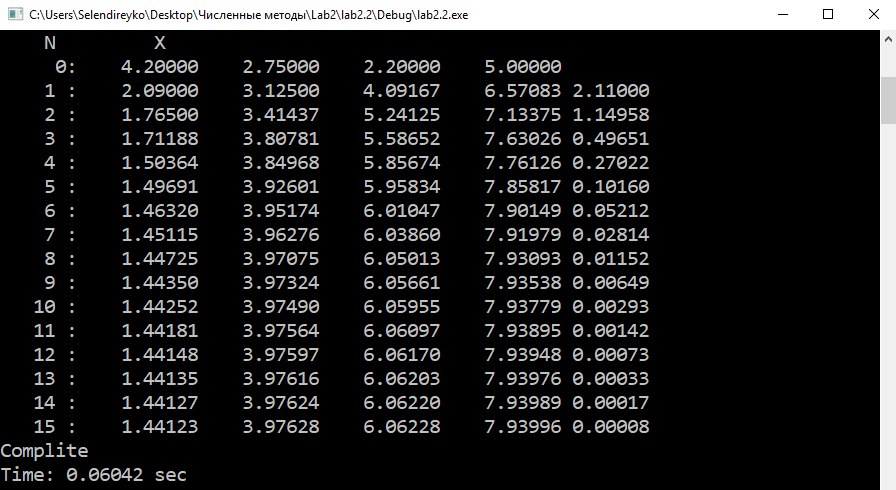
Модули диагональных элементов матрицы *А* для каждой строки больше суммы остальных элементов строки, следовательно метод Зейделя будет сходиться независимо от выбора начального приближения.

Воспользуемся разработанной программой и решим систему методом Халецкого и методом Зейделя.

Вывод программы, при решении методом Халецкого:



Вывод программы при решении методом Зейделя:



# Приложения

**Исходный код**

**lab2.cpp**

#include <iostream>

#include <vector>

#include <conio.h>

#include <time.h>

#include <fstream>

using namespace std;

void func\_lab2(vector <vector <double> > A, vector <vector <double> > &L,

vector <vector <double> > &U, int n)

{

U = A;

for (int i = 0; i < n; i++)

for (int j = i; j < n; j++)

L[j][i] = U[j][i] / U[i][i];

for (int k = 1; k < n; k++)

{

for (int i = k - 1; i < n; i++)

for (int j = i; j < n; j++)

L[j][i] = U[j][i] / U[i][i];

for (int i = k; i < n; i++)

for (int j = k - 1; j < n; j++)

U[i][j] = U[i][j] - L[i][k - 1] \* U[k - 1][j];

}

}

void proisv(vector <vector <double> > A, vector <vector <double> > B,

vector <vector <double> > &R, int n)

{

for (int i = 0; i < n; i++)

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

for (int k = 0; k < n; k++)

R[i][j] += A[i][k] \* B[k][j];

}

void show(vector <vector <double> > A, int n)

{

for (int i = 0; i < n; i++)

{

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

{

cout << "\t" << A[i][j] << "\t";

}

cout << endl;

}

}

int main()

{

srand(time(0));

int i;

int n = 4;

ifstream in("matrix.txt");

ifstream inB("B.txt");

vector <vector <double> > A(n), L(n), U(n), R(n);

double ar[n\*n];

double B[n];

for (int i = 0; i<n\*n; i++)

in >> ar[i];

for (int i = 0; i<n; i++)

inB >> B[i];

for (int i = 0; i < n; i++)

{

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

{

A[i].push\_back(ar[i\*n + j]);

L[i].push\_back(0);

U[i].push\_back(0);

R[i].push\_back(0);

}

}

func\_lab2 (A, L, U, n);

\_getch();

cout << endl;

cout << "B:" << endl;

for (int i = 0; i<n; i++)

cout << B[i] << "\t";

cout << endl << endl;

double Y[n];

Y[0] = B[0] / L[0][0];

for (int i = 1; i<n; i++) {

Y[i] = B[i];

for (int j = 0; j<i; j++) {

Y[i] = Y[i] - Y[j] \* L[i][j];

}

Y[i] = Y[i] / L[i][i];

}

cout << "Y:" << endl;

for (int i = 0; i<n; i++)

cout << Y[i] << "\t";

cout << endl << endl;

double X[n];

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

X[i] = Y[i];

for (int j = n - 1; j>i; j--) {

X[i] = X[i] - X[j] \* U[i][j];

}

X[i] = X[i] / U[i][i];

}

cout << "X:" << endl;

for (int i = 0; i<n; i++)

cout << X[i] << "\t";

cout << endl << endl;

double O[n];

cout << "AX = " << endl;

for (int i = 0; i<n; i++)

O[i] = 0;

for (int i = 0; i<n; i++) {

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

O[i] += A[i][j] \* X[j];

}

cout << O[i] << "\t";

}

cout << endl;

system("pause");

return 0;

}

**lab2.2.cpp**

#include <iostream>

#include <cmath>

#include <stdlib.h>

#include <iomanip>

#include <ctime>

#include <ratio>

#include <chrono>

using namespace std;

using namespace std::chrono;

class Matrix

{

private:

double\*\* matrix;

size\_t x\_size;

size\_t y\_size;

public:

Matrix(int x, int y) : x\_size(x), y\_size(y)

{

matrix = new double\*[x\_size];

for (int i = 0; i < x\_size; ++i)

matrix[i] = new double[y\_size];

for (int i = 0; i < x\_size; ++i)

{

for (int j = 0; j < y\_size; ++j)

{

matrix[i][j] = 0;

}

}

}

Matrix(const Matrix& m)

{

matrix = m.matrix;

x\_size = m.x\_size;

y\_size = m.y\_size;

}

~Matrix()

{

for (int j = 0; j < y\_size; ++j)

delete[]matrix[j];

delete[]matrix;

}

void fillMatrix()

{

if (x\_size == 0 && y\_size == 0)

{

cerr << "ERROR net matix, vvedite new matrix";

exit(0);

}

cout << "Vvedite matrix" << x\_size << "x" << y\_size << endl;

for (int i = 0; i < x\_size; ++i)

{

for (int j = 0; j < y\_size; ++j)

{

cin >> matrix[i][j];

}

}

}

void printMatrix()const

{

cout << "Matrix " << x\_size << "x" << y\_size << endl;

cout.setf(ios::fixed);

cout.precision(3);

for (int i = 0; i < x\_size; i++)

{

for (int j = 0; j < y\_size; j++)

{

cout << matrix[i][j] << " ";

}

cout << endl;

}

}

void zeidel(double\* vec)

{

const size\_t MAX = 100;

double\* iterVector = nullptr;

iterVector = new double[x\_size];

for (size\_t i = 0; i < x\_size; i++)

{

iterVector[i] = vec[i];

}

double\*\* vector = nullptr;

vector = new double\*[x\_size];

for (size\_t i = 0; i < x\_size; i++)

{

vector[i] = new double[x\_size];

}

for (size\_t i = 0; i < x\_size; i++)

for (size\_t j = 0; j < x\_size; j++)

{

vector[i][j] = 0;

}

int n = 0;

for (size\_t i = 0; i < x\_size; i++)

{

for (size\_t j = 0; j < x\_size; j++)

{

if (i == j)

{

vector[i][j] = 0;

}

else vector[i][j] = -matrix[i][j] / matrix[i][n];

}

iterVector[i] = vec[i] / matrix[i][n];

n++;

}

cout << setw(15) << "\tNew Matrix A" << setw(15) << "\t\tVector free chlenov В" << endl;

for (size\_t i = 0; i < x\_size; i++) {

for (size\_t j = 0; j < x\_size; j++)

{

cout << setw(8) << vector[i][j] << " ";

}

cout << setw(8) << " " << iterVector[i];

cout << endl;

}

cout << endl << endl;

double\*\* mat = nullptr;

mat = new double\*[MAX];

for (size\_t j = 0; j < MAX; j++)

{

mat[j] = new double[x\_size];

}

for (size\_t i = 0; i < MAX; i++)

{

for (size\_t j = 0; j < x\_size; j++)

mat[i][j] = 0;

}

cout.setf(ios::fixed);

cout.precision(5);

double max = 0;

for (size\_t i = 0; i < x\_size; ++i)

{

mat[0][i] = iterVector[i];

}

cout << setw(5) << "N" << setw(10) << "X" << endl;

cout << " 0: ";

for (size\_t i = 0; i < x\_size; ++i)

cout << setw(10) << mat[0][i] << " ";

cout << endl;

for (size\_t i = 1; ; ++i)

{

cout << setw(5) << i << " : ";

for (size\_t j = 0; j < x\_size; ++j)

{

mat[i][j] += iterVector[j];

for (size\_t k = 0; k < x\_size; ++k)

{

mat[i][j] += vector[j][k] \* mat[i - 1][k];

}

//iterVector[j] = mat[i][j];

double k = fabs(mat[i][j] - mat[i - 1][j]);

if (k > max) max = k;

cout << setw(10) << mat[i][j] << " ";

}

cout << setw(6) << max;

if (max < 0.0001)

{

cout << endl << "Complite" << endl;

break;

}

else max = 0;

cout << endl;

}

for (size\_t i = 0; i < MAX; ++i)

{

delete[]mat[i];

}

delete[]mat;

for (size\_t i = 0; i < x\_size; ++i)

{

delete[]vector[i];

}

delete[]vector;

delete[]iterVector;

}

};

int main()

{

setlocale(LC\_CTYPE, "");

size\_t M;

cout << "Metod Zeidelya" << endl << endl;

cout << "Kol-vo yravneniy: "; cin >> M;

cout << "Svobodnie chleni: (" << M << "): ";

double\* vec;

vec = new double[M];

for (size\_t i = 0; i < M; ++i) cin >> vec[i];

Matrix matrix(M, M);

matrix.fillMatrix();

steady\_clock::time\_point zeidel\_time1 = steady\_clock::now();

matrix.zeidel(vec);

steady\_clock::time\_point zeidel\_time2 = steady\_clock::now();

duration<double> zeidel\_span = duration\_cast<duration<double>>(zeidel\_time2 - zeidel\_time1);

//cout << "Check:\n 5 1 -1 2 2.97525 11.99996\n 0.5 -4 2 -1 X 0.55830 = 0.499955\n-1 0.2 -6 3 -0.31451 -6.60018\n 1 -0.5 3 -6 -1.87455 12.99987" << endl;

cout << "Time: " << zeidel\_span.count() << " sec" << endl;

system("Pause");

return 0;

}