Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Кафедра ИС

Отчет

по лабораторной работе № 5(4)

«Исследование прямых методов

решения систем линейных уравнений»

Выполнил

ст. гр. И12д

Серегин А.В.

Проверил:

асс. Дрозин А. Ю.

Севастополь

2015

1.Цель работы

Выполнение лабораторной работы имеет целью формирование навыков практических расчетов при решении систем линейных уравнений. В данной работе необходимо изучить метод определителей и различные модификации метода исключения Гаусса.

2.Варианты задания

Вариант 6:

1. Решить систему , по методу определителей.

2. Решить систему , методом Гаусса.

3. Решить систему , методом Холецкого.

4. Решить систему , методом Жордана.

Расчеты вести с 8 знаками после запятой.

3. Код программы

Ниже представлен код программы выполняющей расчеты соответствующие варианту задания.

#include <iostream>

#include <iomanip>

#include "Matrix.h"

std::vector<double> Kramer(const Matrix<double>& \_A,

const std::vector<double>& \_b)

{

double D = \_A.determinant();

size\_t n = \_A.heigh();

Matrix<double> A;

std::vector<double> answer;

std::vector<double> b(\_b);

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

{

A = \_A;

A.set\_col(i, b);

answer.push\_back(A.determinant() / D);

}

return answer;

}

std::vector<double> Gauss(Matrix<double> A,

const std::vector<double>& b)

{

size\_t n = A.heigh();

std::vector<double> answer(n);

double temp;

A.push\_right(b);

for (size\_t i = 0; i < n - 1; i++)

for (size\_t j = i + 1; j < n; j++)

for (size\_t k = n; k > i; k--)

A[j][k] = A[i][k] \* A[j][i] - A[j][k] \* A[i][i];

for (size\_t i = n - 1; i != size\_t(-1); i--)

{

temp = 0;

for (size\_t j = i + 1; j < n; j++)

temp += answer[j] \* A[i][j];

answer[i] = (A[i][n] - temp) / A[i][i];

}

return answer;

}

std::vector<double> Cholesky(const Matrix<double>& \_A,

const std::vector<double>& \_b)

{

size\_t n = \_A.heigh();

Matrix<double> B;

Matrix<double> C;

\_A.choleskyBC(B, C);

std::vector<double> y(3);

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

{

y[i] = \_b[i];

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

y[i] -= B[i][j] \* y[j];

y[i] /= B[i][i];

}

std::vector<double> answer(n);

for (size\_t i = n-1; i != size\_t(-1); i--)

{

answer[i] = y[i];

for (size\_t j = i + 1; j < n; j++)

answer[i] -= C[i][j] \* answer[j];

}

return answer;

}

std::vector<double> Gauss\_Jordan(Matrix<double> A,

const std::vector<double>& \_b)

{

size\_t n = A.heigh();

std::vector<double> answer;

A.push\_right(\_b);

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

{

for (size\_t j = i + 1; j < n; j++)

for (size\_t k = n; k > i; k--)

A[j][k] = A[i][k] \* A[j][i] - A[j][k] \* A[i][i];

for (size\_t j = i - 1; j != size\_t(-1); j--)

for (size\_t k = n; k != size\_t(-1); k--)

A[j][k] = A[i][k] \* A[j][i] - A[j][k] \* A[i][i];

}

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

answer.push\_back(A[i][n] / A[i][i]);

return answer;

}

int main()

{

setlocale(LC\_ALL, "Russian");

Matrix<double> A =

{

{ 2.1, -4.5, -2.0 },

{ 3.0, 2.5, 4.3 },

{ -6.0, 3.5, 2.5}

};

std::vector<double> b = { 18.07, 4.21, -18.25 };

std::cout.setf(std::ios::fixed);

std::cout.precision(2);

for (size\_t i = 0; i < 3; i++)

{

std::cout << '|';

for (size\_t j = 0; j < 3; j++)

std::cout<< std::setw(7) << A[i][j] <<' ';

std::cout << "| |" << std::setw(7) << b[i] << " |" << std::endl;

}

std::cout.precision(8);

std::cout << "Метод Крамера:" << std::endl;

std::vector<double> answer = Kramer(A, b);

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

std::cout << "X" << i << " = " << answer[i] << std::endl;

std::cout << std::endl;

std::cout << "Метод Гаусса:" << std::endl;

answer = Gauss(A, b);

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

std::cout << "X" << i << " = " << answer[i] << std::endl;

std::cout << std::endl;

std::cout << "Метод Холецкого:" << std::endl;

answer = Cholesky(A, b);

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

std::cout << "X" << i << " = " << answer[i] << std::endl;

std::cout << std::endl;

std::cout << "Метод Жордана:" << std::endl;

answer = Gauss\_Jordan(A, b);

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

std::cout << "X" << i << " = " << answer[i] << std::endl;

system("pause");

return 0;

}

Ниже представлен метод класса Matrix модуля Matrix.h для реализации разложения Холецкого.

template< class T >

void Matrix<T>::choleskyBC(Matrix<T>& B, Matrix<T>& C) const

{

if (nRow\_ == nCol\_)

{

B.resize(nRow\_, nRow\_);

C.resize(nCol\_, nCol\_);

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

{

for (size\_t j = i; j < nRow\_; j++)

{

B[j][i] = matrix\_[j][i];

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

B[j][i] -= B[j][k] \* C[k][i];

}

for (size\_t j = i; j < nRow\_; j++)

{

C[i][j] = matrix\_[i][j];

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

C[i][j] -= B[i][k] \* C[k][j];

C[i][j] /= B[i][i];

}

}

}

else

throw bad\_Matrix("wrong size");

}

4. Результаты выполнения

На рисунке ниже представлены результаты выполнения программы.

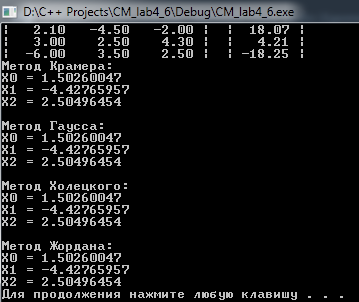


Рисунок 1 – Вывод корней системы.

Выводы

В ходе лабораторной работы мы научились решать системы линейных уравнений различных размеров. Изучили методы Гаусса, Крамера, Жордана и Холецкого.