**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**"Уфимский государственный авиационный технический университет"**

**Кафедра** Высокопроизводительных вычислительных технологий и систем

**Дисциплина:** Численные методы

**Отчет по лабораторной работе № 3**

**Тема:** «Прямые методы решения

систем линейных алгебраических уравнений»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Группа ПМ-353 | Фамилия И.О. | Подпись | Дата | Оценка |
| Студент | Шамаев И.Р. |  |  |  |
| Принял |  |  |  |  |

**Уфа 2021**

**Цель:** получить навык численного решения систем линейных алгебраических уравнений (СЛАУ) с использованием различных прямых методов.

**Теоретическая часть**

***Решение СЛАУ методом Гаусса с выбором ведущего элемента***

Основное увеличение ошибки в методе происходит во время прямого хода, когда ведущая k-я строка умножается на коэффициенты

Если коэффициенты> *1*, то ошибки, полученные на предыдущих шагах, накапливаются. Чтобы этого избежать, применяется модификация метода Гаусса с выбором главного элемента. На каждом шаге к обычной схеме добавляется выбор максимального элемента по столбцу следующим образом:

Пусть по ходу исключения неизвестных получена система уравнений:

Найдем такое *l*, что и поменяем местами *k*-е и *l*-е уравнения.

Такое преобразование во многих случаях существенно уменьшает чувствительность решения к погрешностям округления при вычислениях.

***Решение СЛАУ методом LU-разложения***

LU-разложение — представление [матрицы](http://ru.math.wikia.com/wiki/%D0%9C%D0%B0%D1%82%D1%80%D0%B8%D1%86%D0%B0?veaction=edit&redlink=1) A в виде LU, где L — [нижнетреугольная матрица](http://ru.math.wikia.com/wiki/%D0%A2%D1%80%D0%B5%D1%83%D0%B3%D0%BE%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%BC%D0%B0%D1%82%D1%80%D0%B8%D1%86%D0%B0?action=edit&redlink=1" \o "Треугольная матрица (страница не существует)) с единичной диагональю, а U — верхнетреугольная.

Матрица L является нижнетреугольной с единичной диагональю, поэтому ее [определитель](http://ru.math.wikia.com/wiki/%D0%9E%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C) равен 1. Матрица U — верхнетреугольная матрица, значит ее определитель равен произведению элементов, расположенных на главной диагонали.

Будем использовать следующие обозначения для элементов матриц ; причем диагональные элементы матрицы  Тогда, если известно LU-разложение матрицы, её определитель можно вычислить по формуле

Найти матрицы L и U можно следующим образом (выполнять шаги следует строго по порядку, т.к. следующие элементы находятся с использованием предыдущих):

Для i=2…n:

В итоге мы получим матрицы — L и U.

Полученное LU-разложение матрицы A (матрица коэффициентов системы) может быть использовано для решения семейства систем линейных уравнений с различными векторами b в правой части:

Если известно LU-разложение матрицы A, A=LU, исходная система может быть записана как

Эта система может быть решена в два шага. На первом шаге решается система

Поскольку *L* — нижняя треугольная матрица, эта система решается непосредственно прямой подстановкой. На втором шаге решается система

И поскольку U — верхняя треугольная матрица, эта система решается непосредственно обратной подстановкой.

***Решение СЛАУ с симметричной матрицей методом квадратного корня***

Матрица A представляется в виде

где S – правая треугольная матрица, S\*– сопряженная с ней, т.е.

причем все , D – диагональная матрица с элементами *dii ,* равными +1 или -1. Матричное равенство представления матрицы А образует уравнений

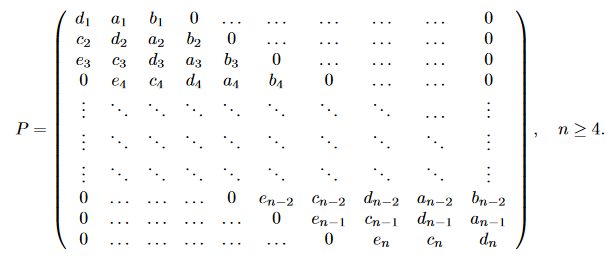
Аналогичные уравнения при отброшены, так как уравнения, соответствующие парам эквивалентны. Отсюда получаем рекуррентные формулы для определения элементов *dii* и *sij:*

Матрица S является правой треугольной, и, таким образом, после получения представления для матрицы А решение исходной системы также сводится к последовательному решению двух систем с треугольными матрицами. Заметим, что в случае

***Решение методом прогонки СЛАУ с пяти диагональной матрицей***

Пяти диагональная линейная система имеет форму

где P – матрица размерности N вида:



и .

Решение пяти диагональной матрицы методом прогонки выводится аналогично методу прогонки для трех диагональной матрицы.

**Задание №1**

Задание: написать вычислительную программу для решения СЛАУ методом Гаусса с выбором ведущего элемента, решить задачу о рациональной интерполяции.

Описание: реализованная программа позволяет решить систему линейных алгебраических уравнений методом Гаусса с выбором ведущего элемента в столбце или строке.

Результат: согласно варианту, составляется СЛАУ. СЛАУ решается методом Гаусса.

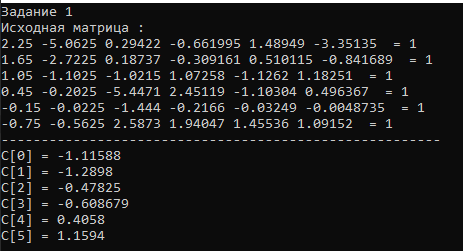


Рисунок 1. Реализация программы

Рисунок 2. График интерполирующего многочлена

**Задание №2**

Задание: написать программу для решения СЛАУ методом LU-разложения.

Описание: в ходе выполнения программы выводится изначальная матрица; далее выводятся матрицы *L* и *U*, полученных в результате работы метода; далее выводятся найденное решение и результат проверки.

Результат:

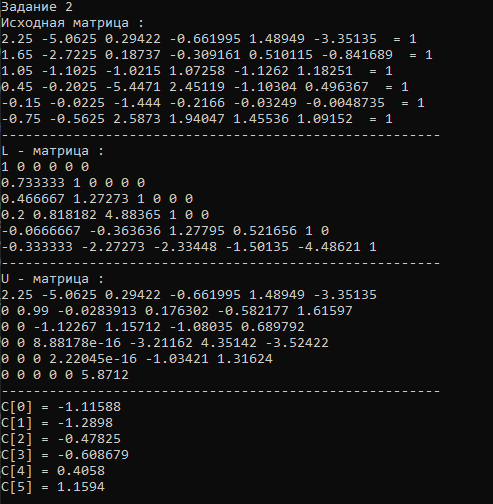


Рисунок 3. Пример реализации программы

**Задание №3**

Задание: написать вычислительную программу на языке программирования C++ для решения СЛАУ с симметричной матрицей методом квадратного корня. С использованием написанной программы решить задачу об аппроксимации функции из первой лабораторной работы, заданной на равномерной сетке из 20 узлов, многочленами степени 1≤𝑛≤12 с использованием метода наименьших квадратов.

Описание:

Результат:

**Задание №4**

Задание: написать программу для решения методом прогонки СЛАУ с 5-диагональной матрицей.

Описание:

Результат:

**Заключение**

В ходе проделанной лабораторной работы был изучен теоретический материал необходимый для решения поставленных задач по численному решению систем линейных алгебраических уравнений с использованием различных прямых методов и получен навык проведения вычислительного эксперимента, направленного на их решение.

Для каждой поставленной задачи написана вычислительная программа на языке программирования С++, выполняющая необходимые построения и расчеты по нахождению решения системы линейных алгебраических уравнений.

**Приложение А**

# *Листинг программы для задания 1:*

#include <iostream>

#include <vector>

#include <fstream>

const int n = 6;

using namespace std;

vector<double> gauss(vector<vector<double>> k, vector<double> R);

vector<double> gauss(vector <vector<double>> k, vector<double> R)

{

double max;

vector<double> x;

x.resize(n);

int c, index;

const double eps = 0.00001;

c = 0;

while (c < n)

{

max = abs(k[c][c]);

index = c;

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

{

if (abs(k[i][c]) > max)

{

max = abs(k[i][c]);

index = i;

}

}

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

{

double temp = k[c][j];

k[c][j] = k[index][j];

k[index][j] = temp;

}

double temp = R[c];

R[c] = R[index];

R[index] = temp;

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

{

double temp = k[i][c];

if (abs(temp) < eps) continue;

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

k[i][j] = k[i][j] / temp;

R[i] = R[i] / temp;

if (i == c) continue;

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

k[i][j] = k[i][j] - k[c][j];

R[i] = R[i] - R[c];

}

c++;

}

for (c = n - 1; c >= 0; c--)

{

x[c] = R[c];

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

R[i] = R[i] - k[i][c] \* x[c];

}

return x;

}

int main()

{

setlocale(LC\_ALL, "Russian");

vector<double> C;

vector<double> x{ -2.25 , -1.65, -1.05, -0.45, 0.15, 0.75 };

vector<double> y{ 0.29422, 0.18737, -1.0215, -5.4471, -1.4440, 2.5873 };

vector<double> R{ 1,1,1,1,1,1 };

vector<vector<double>> A;

A.resize(n);

cout << "Задание 1 " << endl;

cout << "Исходная матрица : " << endl;

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

{

A[i] = { -x[i], -pow(x[i],2), y[i], x[i] \* y[i], y[i] \* pow(x[i], 2), y[i] \* pow(x[i], 3) };

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

{

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

}

cout << " = " << R[i] << endl;

}

cout << "-------------------------------------------------------" << endl;

C = gauss(A, R);

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

{

cout << "C[" << i << "] = " << C[i] << endl;

}

ofstream fout;

fout.open("Result.txt");

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

{

fout << x[i] << " " << y[i] << endl;

}

fout.close();

ofstream out;

out.open("Result1.txt");

out << 1 << endl;

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

{

out << C[i] << endl;

}

out.close();

return 0;

}

*Листинг программы для задания 2*

#include <iostream>

#include <vector>

#include <fstream>

const int n = 6;

using namespace std;

vector<double> Gauss(vector <vector<double>> k, vector<double> R)

{

double max;

vector<double> x;

x.resize(n);

int c, index;

const double eps = 0.00001;

c = 0;

while (c < n)

{

max = abs(k[c][c]);

index = c;

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

{

if (abs(k[i][c]) > max)

{

max = abs(k[i][c]);

index = i;

}

}

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

{

double temp = k[c][j];

k[c][j] = k[index][j];

k[index][j] = temp;

}

double temp = R[c];

R[c] = R[index];

R[index] = temp;

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

{

double temp = k[i][c];

if (abs(temp) < eps) continue;

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

k[i][j] = k[i][j] / temp;

R[i] = R[i] / temp;

if (i == c) continue;

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

k[i][j] = k[i][j] - k[c][j];

R[i] = R[i] - R[c];

}

c++;

}

for (c = n - 1; c >= 0; c--)

{

x[c] = R[c];

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

R[i] = R[i] - k[i][c] \* x[c];

}

return x;

}

void Print(vector<vector<double>> A)

{

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

{

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

{

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

}

cout << endl;

}

cout << "-------------------------------------------------------" << endl;

}

vector<vector<double>> LU(vector<vector<double>> k)

{

vector<vector<double>> v;

v.resize(n);

for (int i = 0; i < n; i++) v[i].resize(n \* 2 + 1);

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

{

double max = 0;

max = k[i][i];

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

{

v[j][i] = k[j][i] / max;

}

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

{

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

{

k[j][z] -= k[i][z] \* v[j][i];

}

}

}

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

{

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

{

v[i][j + n] = k[i][j];

}

}

return v;

}

void Result(vector<double> C, vector<double> x, vector<double> y)

{

ofstream fout;

fout.open("Result.txt");

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

{

fout << x[i] << " " << y[i] << endl;

}

fout.close();

ofstream out;

out.open("Result1.txt");

out << 1 << endl;

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

{

out << C[i] << endl;

}

out.close();

}

int main()

{

setlocale(LC\_ALL, "Russian");

vector<double> C; vector<double> Y;

vector<double> x{ -2.25 , -1.65, -1.05, -0.45, 0.15, 0.75 };

vector<double> y{ 0.29422, 0.18737, -1.0215, -5.4471, -1.4440, 2.5873 };

vector<double> R{ 1,1,1,1,1,1 };

vector<vector<double>> A; A.resize(n);

vector<vector<double>> lu;

cout << "Задание 2" << endl;

cout << "Исходная матрица :" << endl;

vector<vector<double>> L; vector<vector<double>> U;

L.resize(n); U.resize(n);

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

{

L[i].resize(n); U[i].resize(n);

}

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

{

A[i] = { -x[i], -pow(x[i],2), y[i], x[i] \* y[i], y[i] \* pow(x[i], 2), y[i] \* pow(x[i], 3) };

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

{

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

}

cout << " = " << R[i] << endl;

}

cout << "-------------------------------------------------------" << endl;

lu = LU(A);

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

{

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

{

L[i][j] = lu[i][j];

}

for (int j = n; j < 2 \* n; j++)

{

U[i][j - n] = lu[i][j];

}

}

cout << "L - матрица :" << endl;

Print(L);

cout << "U - матрица :" << endl;

Print(U);

Y = Gauss(L, R);

C = Gauss(U, Y);

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

{

cout << "C[" << i << "] = " << C[i] << endl;

}

Result(C, x, y);

return 0;

}