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

$$\frac{a_{i,k}}{a_{k,k}}, i=k+1,\ldots,n.$$

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:

$$x_i+\sum_{j=i+1}^{n} a_{ij}^{i} x^{j}=b_i^{i}, i=1,\ldots,k-1,$$

$$\sum_{j=k}^{n} a_{ij}^{k-1} x_j=b_i^{k-1}, i=k,\ldots,n.$$

Найдем такое l, что $\left|a_{l,k}{}^{k-1}\right|=\max_{j=k,\ldots,n}\left|a_{j,k}{}^{k-1}\right|$ 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 A в виде LU, где L — нижнетреугольная матрица с единичной диагональю, а U — верхнетреугольная.

Матрица L является нижнетреугольной с единичной диагональю, поэтому ее определитель равен 1. Матрица U — верхнетреугольная матрица, значит ее определитель равен произведению элементов, расположенных на главной диагонали.

Будем использовать следующие обозначения для элементов матриц $L=(l_{ij}),U=(u_{ij}),i,j=1\ldots n$; причем диагональные элементы матрицы $L: l_{ii}=1, i=1\ldots n.$ Тогда, если известно LU-разложение матрицы, её определитель можно вычислить по формуле $det(A)=det(LU)=det(L)det(U)=det(U)$

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

$$u_{1j}=a_{1j}, j=1\ldots n,$$

$$l_{j1}=\frac{a_{j1}}{u_{11}}, j=2\ldots n\,(u_{11}\neq 0)$$

Для i=2…n:

$$u_{ij}=a_{ij}-\sum_{k=1}^{i} l_{ik}u_{kj},\ j=i\ldots n$$

$$l_{ji}=\frac{1}{u_{ii}}\left(a_{ji}-\sum_{k=1}^{i} l_{jk}u_{ki}\right),\ j=i+1\ldots n$$

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:

$$Ax=b$$

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

$$LUx=b.$$

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

$$Ly=b.$$

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

$$Ux=y.$$

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

$$A=S^{¿}DS,$$

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

$$S=\begin{pmatrix} s_{11} & s_{12} & \ldots \\ 0 & s_{22} & \ldots \\ \ldots & \ldots & \ldots \end{pmatrix},$$

причем все $s_{ii}>0$, D – диагональная матрица с элементами d_{ii}, равными +1 или -1. Матричное равенство представления матрицы A образует *систему* уравнений

$$a_{ij}=\sum_{k=1}^{i}\bar{s}_{ki}s_{kj}d_{kk}=¿\bar{s}_{1i}s_{1j}d_{11}+\ldots+\bar{s}_{ii}s_{ij}d_{ii}\,\text{при}\,i\le j.¿$$

Аналогичные уравнения при $i>j$ отброшены, так как уравнения, соответствующие парам $(i,j)u(j,i)$, эквивалентны. Отсюда получаем рекуррентные формулы для определения элементов d_{ii} и s_{ij}:

$$d_{ii}=sign\left(a_{ii}-\sum_{k=1}^{i-1}|s_{ki}|^2 d_{kk}\right), s_{ii}=\sqrt{\left|a_{ii}-\sum_{k=1}^{i-1}|s_{ki}|^2 d_{kk}\right|},$$

$$s_{ij}=(a¿¿ij-\sum_{k=1}^{i-1}\bar{s}_{ki}s_{kj}d_{kk})/(s_{ii}d_{ii})\,\text{при}\,i<j.¿$$

Матрица S является правой треугольной, и, таким образом, после получения представления для матрицы A решение исходной системы также сводится к последовательному решению двух систем с треугольными матрицами. Заметим, что в случае $A>0\,все\,d_{ii}=1\,и\,A=S^{¿}S$.

Решение методом прогонки СЛАУ с пяти диагональной матрицей

Пяти диагональная линейная система имеет форму

$$PX=Y,$$

где P – матрица размерности N вида:

$$P=\begin{pmatrix} d_1 & a_1 & b_1 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ c_2 & d_2 & a_2 & b_2 & 0 & \dots & \dots & \dots & \dots & 0 \\ e_3 & c_3 & d_3 & a_3 & b_3 & 0 & \dots & \dots & \dots & 0 \\ 0 & e_4 & c_4 & d_4 & a_4 & b_4 & 0 & \dots & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & \dots & 0 & e_{n-2} & c_{n-2} & d_{n-2} & a_{n-2} & b_{n-2} \\ 0 & \dots & \dots & \dots & \dots & 0 & e_{n-1} & c_{n-1} & d_{n-1} & a_{n-1} \\ 0 & \dots & \dots & \dots & \dots & \dots & 0 & e_n & c_n & d_n \end{pmatrix}, \quad n \geq 4.$$

и $X=(x_1,x_2,\dots,x_n)^t, Y=(y_1,y_2,\dots,y_n)^t$.

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

Задание №1

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

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

<u>Результат:</u> согласно варианту, составляется СЛАУ. СЛАУ решается методом Гаусса.

```
Задание 1
Исходная матрица :
2.25 -5.0625 0.29422 -0.661995 1.48949 -3.35135  = 1
1.65 -2.7225 0.18737 -0.309161 0.510115 -0.841689  = 1
1.05 -1.1025 -1.0215 1.07258 -1.1262 1.18251  = 1
0.45 -0.2025 -5.4471 2.45119 -1.10304 0.496367  = 1
-0.15 -0.0225 -1.444 -0.2166 -0.03249 -0.0048735  = 1
-0.75 -0.5625 2.5873 1.94047 1.45536 1.09152  = 1
-------------------------------------------------------
C[0] = -1.11588
C[1] = -1.2898
C[2] = -0.47825
C[3] = -0.608679
C[4] = 0.4058
C[5] = 1.1594
```

Рисунок 1. Реализация программы

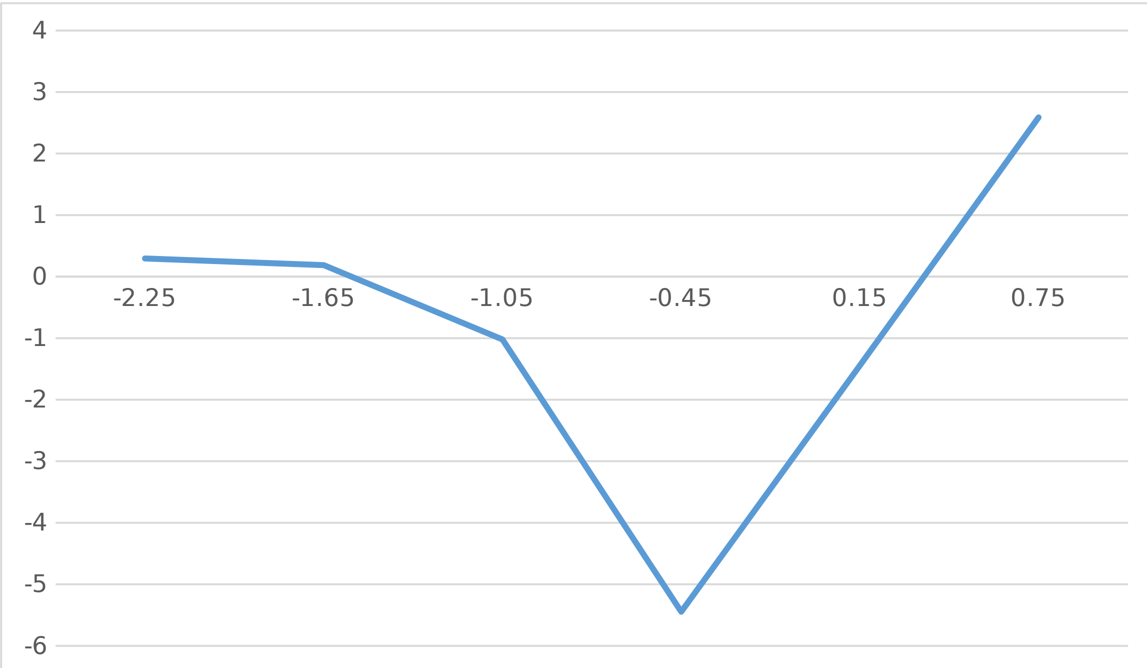

Рисунок 2. График интерполирующего многочлена

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:

```
Задание 2
Исходная матрица :
2.25 -5.0625 0.29422 -0.661995 1.48949 -3.35135  = 1
1.65 -2.7225 0.18737 -0.309161 0.510115 -0.841689  = 1
1.05 -1.1025 -1.0215 1.07258 -1.1262 1.18251  = 1
0.45 -0.2025 -5.4471 2.45119 -1.10304 0.496367  = 1
-0.15 -0.0225 -1.444 -0.2166 -0.03249 -0.0048735  = 1
-0.75 -0.5625 2.5873 1.94047 1.45536 1.09152  = 1
-------------------------------------------------------
L - матрица :
1 0 0 0 0 0
0.733333 1 0 0 0 0
0.466667 1.27273 1 0 0 0
0.2 0.818182 4.88365 1 0 0
-0.0666667 -0.363636 1.27795 0.521656 1 0
-0.333333 -2.27273 -2.33448 -1.50135 -4.48621 1
-------------------------------------------------------
U - матрица :
2.25 -5.0625 0.29422 -0.661995 1.48949 -3.35135
0 0.99 -0.0283913 0.176302 -0.582177 1.61597
0 0 -1.12267 1.15712 -1.08035 0.689792
0 0 8.88178e-16 -3.21162 4.35142 -3.52422
0 0 0 2.22045e-16 -1.03421 1.31624
0 0 0 0 0 5.8712
-------------------------------------------------------
C[0] = -1.11588
C[1] = -1.2898
C[2] = -0.47825
C[3] = -0.608679
C[4] = 0.4058
C[5] = 1.1594
```

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 $1 \leq n \leq 12$ 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 C++, 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;

}
```