МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тихоокеанский государственный университет»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Решение системы линейных алгебраических уравнений

Лабораторная работа №2

по дисциплине «Вычислительная математика»

Выполнил студент	Чекулаев В. Ю.
Факультет, группа	ФКФН, ПО(аб)-81
Проверил	Резак Е.В.

Хабаровск – 2020г.

Задание: Дана система линейных уравнений $Ax = b$.

1) Привести систему линейных уравнений к итерационному виду.

2) Исследовать итерационную последовательность на сходимость.

3) Найти решение системы линейных уравнений методом простой итерации с точностью до ε = 0,00001.

4) Найти решение системы линейных уравнений методом Зейделя с точностью до ε = 0,00001.

Вариант 1. $A=\begin{pmatrix} 24{,}41 & 2{,}42 & 3{,}85 \\ 2{,}31 & 31{,}49 & 1{,}52 \\ 3{,}49 & 4{,}85 & 28{,}92 \end{pmatrix}; b=\begin{pmatrix} 30{,}24 \\ 40{,}95 \\ 42{,}81 \end{pmatrix}$

1. **Приведение системы линейных уравнений к итерационному виду**

Уравнения, входящие в систему $Ax = b$, переставляются так, чтобы выполнялось условие диагонального преобладания (для этой же цели можно использовать другие элементарные преобразования) . Затем первое уравнение разрешается относительно x_1, второе — относительно x_2. При этом получается матрица α с нулевыми диагональными элементами.

Таким образом, получаем систему вида $x = \alpha x+b$.

2. **Исследование итерационной последовательности на сходимость**

Теорема о достаточном условии сходимости метода простых итераций:

Метод простых итераций, реализующийся в процессе последовательных приближений, сходится к единственному решению исходной системы $Ax = b$ при любом начальном приближении $x^{(0)}$ со скоростью не медленнее геометрической прогресии, если какая-либо норма матрицы α меньше единицы, т. е. $\|\alpha\|_s<1\left(s\in\left(1,2,3\right)\right)$.

Замечание:

Условия сходимости выполняются, если в матрице A диагональные элементы преобладают.

3. **Метод простых итераций**

1) Исходная задача $Ax = b$ преобразуется к равносильному виду $x = \alpha x + b$, где α — квадратная матрица порядка n; *b* — столбец.

2) Столбец *b* принимается в качестве начального приближения и далее многократно выполняются действия по уточнению решения, согласно рекуррентному соотношению:

$$x^{(k+1)} = \alpha x^{(k)} + b, k = 1,2,3\ldots$$

3) Итерации прерываются при выполнении условия $\left\| x^{(k+1)} - x^{(k)} \right\| < \varepsilon$, где ε — заданная точность.

4. **Метод Зейделя**

Итерации по методу Зейделя отличаются от метода простых итераций тем, что при нахождении i-той компоненты (k+1)-го приближения сразу используются уже найденные компоненты (k+1)-го приближения с меньшими номерами.

$$x^{(k+1)} = L x^{(k+1)} + U x^{k} + b$$

где L и U являются разложением матрицы α на нижнюю и верхнюю треугольную матрицы соответственно.

Ручной расчет

	A	B	C	D	E	F	G	H	I	J	K
1		Исходная матрица					Переход				
2		A		b			C		d		
3	24,41	2,42	3,85	30,24		0	-0,099139697	-0,157722245	1,2388365424		
4	2,31	31,49	1,52	40,95		-0,073356621	0	-0,048269292	1,3004128295		
5	3,49	4,85	28,92	42,81		-0,120677732	-0,167704011	0	1,4802904564		
6											
7						МЕТОД ПРОСТЫХ ИТЕРАЦИЙ					
8							x^k				
9	k	1	2	3	4	5	6	7	8	9	10
10		1,2388365424	0,8764392747	0,9505088074	0,9349227837	0,9383833067	0,9376606369	0,9378189254	0,9377853518	0,9377926167	0,9377910632
11		1,3004128295	1,1380833945	1,1824106738	1,1735521594	1,1754857818	1,1750694313	1,1751582541	1,1751390627	1,1751431666	1,1751422827
12		1,4802904564	1,112706025	1,1836626026	1,1672901969	1,1706566913	1,169914807	1,1700718408	1,1700378429	1,170045113	1,1700435481
13											
14	||x^k-x^k+1||		0,892311134	0,1893533896	0,0408169439	0,0087606399	0,0018809047	0,0004041452	8,676287E-05	1,863881E-05	4,002301E-06
15											
16											
17						МЕТОД ЗЕЙДЕЛЯ					
18		U			L						
19	0	-0,099139697	-0,157722245	0	0	0					
20	0	0	-0,048269292	-0,073356621	0	0					
21	0	0	0	-0,120677732	-0,167704011	0					
22											
23											
24											
25	k	1	2	3	4	5	6	7			
26		1,2388365424	1,238836545	0,943425021	0,937154233	0,937772628	0,937792668	0,937791395			
27		1,3004128295	1,209535978	1,177496837	1,175065631	1,175135759	1,175142554	1,175142466			
28		1,4802904564	1,112706039	1,172603345	1,170210735	1,170039683	1,170043484	1,170043834			
29											
30	||x^k-x^k+1||		0,4584612715	0,387347971	0,011094604	0,000859575	3,0636E-05	1,711E-06			

Листинг

```
#include<iostream>
#include<fstream>
#include<cmath>
using namespace std;

float* multM(float* M[], float* V, int N){
    float* buf = new float[N];
    for(int i = 0; i < N; i++){
        buf[i] = 0;
    }

    for(int j = 0; j < N; j++){
        for(int i = 0; i < N; i++){
            buf[i] += M[i][j] * V[j];
        }
    }

    return buf;
}

float* sumM(float* V1, float* V2, int N){
    float* buf = new float[N];
    for(int i = 0 ; i < N; i++){
        buf[i] = V1[i] + V2[i];
    }
    return buf;
}

float* sumM(float* V1, float* V2, float* V3, int N){
    float* buf = new float[N];
    for(int i = 0 ; i < N; i++){
        buf[i] = V1[i] + V2[i] + V3[i];
    }
    return buf;
}

float norma(float* V1, float* V2, int N){
    float res = 0;
    for(int i = 0; i < N; i++){
        res += abs(V1[i] - V2[i]);
    }
```

```
    return res;
}

void m1(float* C[], float* d, int N){
    float* x_k = new float[N];
    for(int i = 0; i < N; i++){
        x_k[i] = d[i];
    }
    float* x_k1 = new float[N];
    x_k1 = x_k;
    int k = 0; float E = 0.00001;

    cout << k+1 << ": ";
    for(int i = 0; i < N; i++){
        cout << "x" << i+1 << "=" << x_k1[i] << " ";
    }cout << "\n";
    k++;

    do{
        x_k = x_k1;
        x_k1 = sumM(multM(C, x_k, N), d, N);
        cout << k+1 << ": ";
        for(int i = 0; i < N; i++){
            cout << "x" << i+1 << "=" << x_k1[i] << " ";
        }cout << "\n";
        k++;
    } while(norma(x_k, x_k1, N) > E);

    cout << "\n\n   Ответ:\n   ";
    for(int i = 0 ; i < N; i++){
        cout << "x" << i+1  << "=" << x_k1[i] << " ";
    }cout << "\n";
}

void m2(float* C[], float* d, int N){
    float** L = new float*[N];
    float** U = new float*[N];
    for(int i = 0; i < N; i++){
        L[i] = new float[N];
        U[i] = new float[N];
    }
```

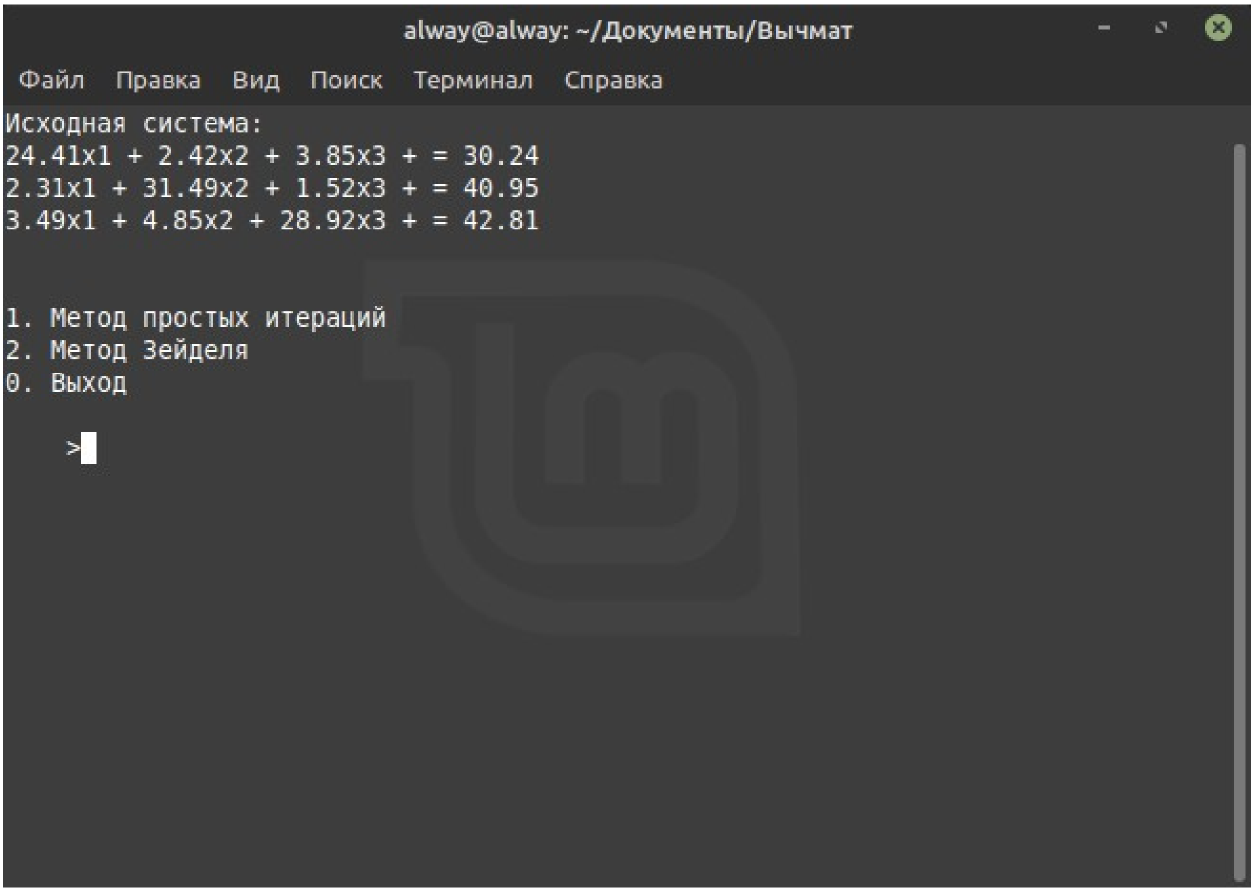
```
    for(int i = 0; i < N; i++){
        for(int j = 0; j < N; j++){
            if(j >= i){
                U[i][j] = C[i][j];
                L[i][j] = 0;
            } else{
                L[i][j] = C[i][j];
                U[i][j] = 0;
            }
        }
    }

    float* x_k = new float[N];
    for(int i = 0; i < N; i++){
        x_k[i] = d[i];
    }
    float* x_k1 = new float[N];
    int k = 0; float E = 0.00001;

    cout << k+1 << ": ";
    for(int i = 0; i < N; i++){
        cout << "x" << i+1 << "=" << x_k[i] << " ";
    }cout << "\n";
    k++;

    do{
        x_k = x_k1;
        x_k1 = sumM(multM(C, x_k, N), d, N);
        x_k1 = sumM(multM(L, x_k1, N), multM(U, x_k, N), d, N);
        cout << k+1 << ": ";
        for(int i = 0; i < N; i++){
            cout << "x" << i+1 << "=" << x_k1[i] << " ";
        }cout << "\n";
        k++;
    }while(norma(x_k, x_k1, N) > E);

    cout << "\n\n   Ответ:\n   ";
    for(int i = 0 ; i < N; i++){
        cout << "x" << i+1  << "=" << x_k1[i] << " ";
    }cout << "\n";
}
```

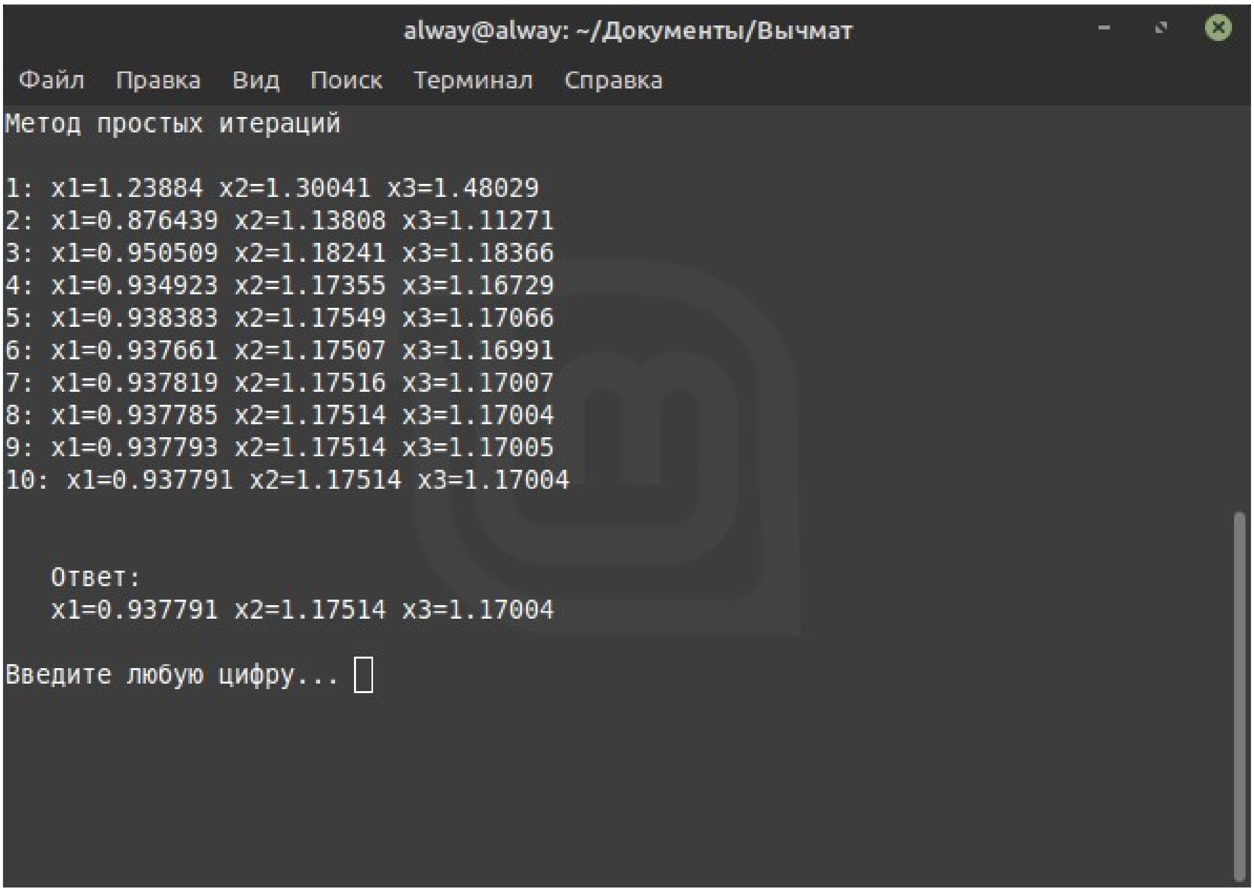
```
void cls(){
    cout << "\033[2J\033[1;1H";
}

int main(){
    int N;
    ifstream file ("matrix.txt");
    if(!file){
        cout << "File is not open!\n";
        return -1;
    }

    file >> N;
    float** A = new float*[N];
    for(int i = 0; i < N; i++){
        A[i] = new float[N];
    }

    for(int i = 0; i < N; i++){
        for(int j = 0; j < N; j++){
            file >> A[i][j];
        }
    }

    float* b = new float[N];

    for(int i = 0; i < N; i++){
        file >> b[i];
    }

    file.close();

    float** C = new float*[N];
    for(int i = 0; i < N; i++){
        C[i] = new float[N];
    }
    for(int i = 0; i < N; i++){
        for(int j = 0; j < N; j++){
            if(i == j){
                C[i][j] = 0;
            } else{
                C[i][j] = -A[i][j]/A[i][i];
```

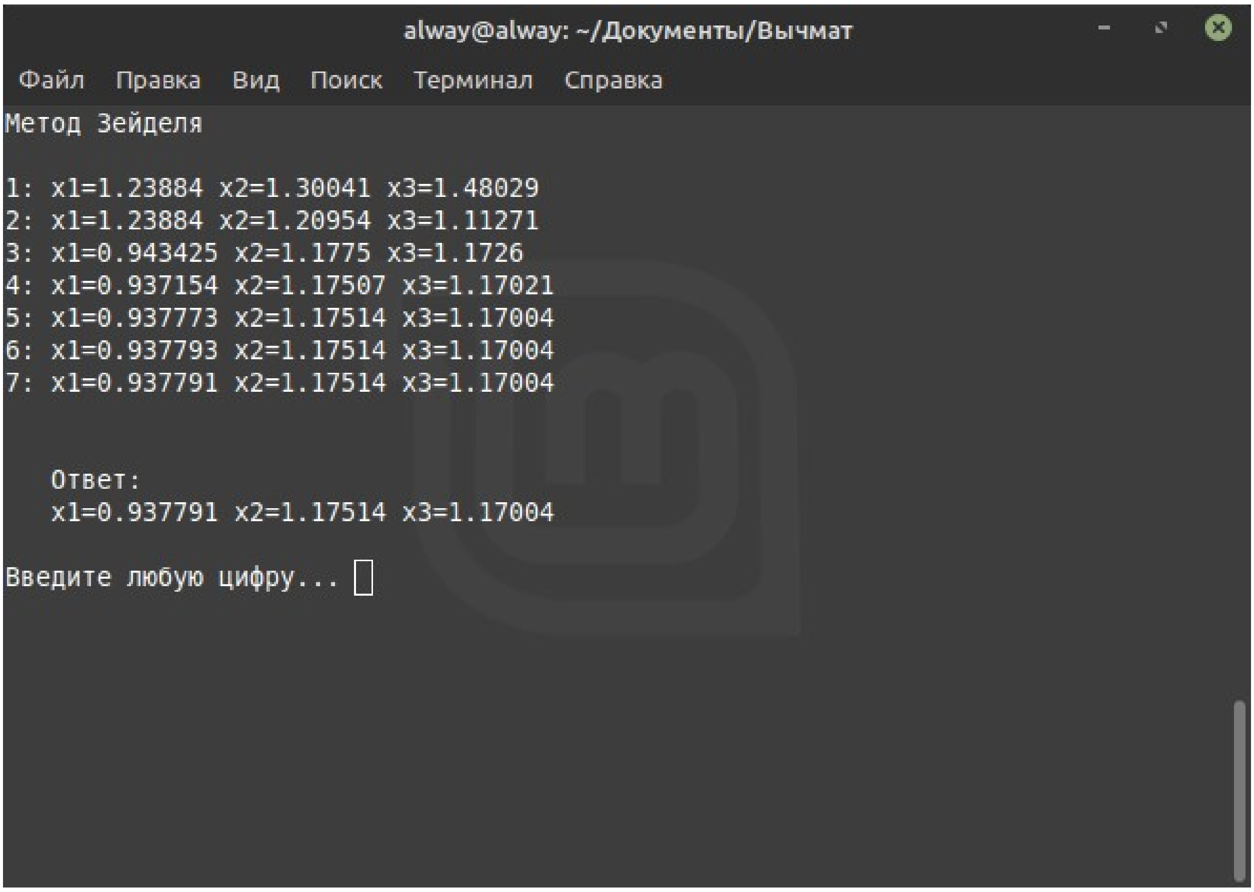
```
        }
    }
}

float* d = new float[N];
for(int i = 0; i < N; i++){
    d[i] = b[i]/A[i][i];
}
int ans;
bool flag = true;
while(flag){
    cls();
    cout << "Исходная система: \n";
    for(int i = 0; i < N; i++){
        for(int j = 0; j < N; j++){
            cout << A[i][j] << "x" << j+1 << " + ";
        }
        cout << "= " << b[i] << "\n";
    }cout << "\n\n";

    cout << "1. Метод простых итераций\n";
    cout << "2. Метод Зейделя\n";
    cout << "0. Выход\n\n    >";
    cin >> ans;

    switch(ans){
        case 1:
            cls();
            cout << "Метод простых итераций\n\n";
            m1(C, d, N);
            cout << "\nВведите любую цифру... ";
            cin >> ans;
            break;
        case 2:
            cls();
            cout << "Метод Зейделя\n\n";
            m2(C, d, N);
            cout << "\nВведите любую цифру... ";
            cin >> ans;
            break;
        case 0:
            flag = false;
```

```
            break;
        default:
            break;
        }
    }

    cls();
    return 0;
}
```

Вывод программы







Вывод

В ходе данной лабораторной работы были изучены два метода решения систем алгебраических линейных уравнений. На основе теоретических данных была написана программа, результаты и количество итераций которой совпали с ручным расчетом. В результате ручных расчетов и работы программы выяснилось, что метод Зейделя более эффективен по сравнению с методом простых итераций.