Группа 4310

Нигамадянов Ф.М.

Маннанов Д.И.

**ЛАБОРАТОРНАЯ РАБОТА №5.**

**ИТЕРАЦИОННЫЕ МЕТОДЫ РЕШЕНИЯ**

**СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ**

Цель работы: научиться решать системы линейных алгебраических уравнений (СЛАУ) методом простых итераций (МПИ) и методом Зейделя с помощью ЭВМ.

Содержание работы:

1. Изучить метод простых итераций и метод Зейделя для решения СЛАУ.

2. На конкретном примере усвоить порядок решения СЛАУ с помощью ЭВМ указанными методами.

3. Составить программу и с ее помощью решить СЛАУ с точностью . Сравнить скорости сходимости метода простых итераций и метода Зейделя.

4. Составить отчет о работе.

Задание (вариант №19).

1. Аналитически решить СЛАУ вида:

(1)

2. Построить рабочие формулы МПИ и метода Зейделя для численного решения системы (1).

3. Составить программу(ы) на любом языке программирования, реализующую(ие) построенные итерационные процессы.

Решение.

1. Аналитическим решением системы являются значения:.

2. Метод простых итераций. Из системы (1) видно, что модули диагональных коэффициентов в каждом уравнении отличны от нуля и больше суммы модулей всех остальных коэффициентов, не считая столбца свободных членов:

. Заметим, что если указанные условия не выполняются, то путем элементарных преобразований систему необходимо к этому виду привести. В нашем случае дополнительных преобразований не требуется.

Разделив каждое уравнение системы (1) на соответствующий диагональный коэффициент, сформируем столбец в левой части и перенесем остальные слагаемые в правую часть и получим рабочие формулы МПИ вида:

(2)

Начальное приближение обычно выбирают равным столбцу свободных членов преобразованной системы . Процесс (2) заканчивается при одновременном выполнении трех условий:

В этом случае значения являются приближенными значениями решения СЛАУ (1).

Метод Зейделя. Более быструю скорость сходимости имеет метод Зейделя, в котором найденное -е приближение сразу же используется для получения -го приближения последующих координат (Рис.1).















Рис.1

Рабочие формулы метода Зейделя запишутся так:

(3)

Условия выхода итерационного процесса (3) и выбор начального приближения аналогичны МПИ.

Листинг программы:

#include <iostream>

#include "locale.h"

#include <math.h>

#include <iomanip>

using namespace std;

const float eps = 0.001; // точность

const float x = 1.0, y = -126.0/705.0, z = - 3.0/4.0; // начальное приближение

const int rr = 10000; // округление

void method\_simple\_implications() // Метод простых итераций

{

int k = 0;

float x0 = x, x1 = x; // x0 = x(k), x1 = x(k+1)

float y0 = y, y1 = y; // y0 = y(k), y1 = y(k+1)

float z0 = z, z1 = z; // z0 = y(k), z1 = z(k+1)

float modx, mody, modz; // modx = |x1 - x0|, mody = |y1 - y0|, modz = |z1 - z0|

do

{

x0 = x1; y0 = y1; z0 = z1;

x1 = x - 0.1 \* y0 - (1.0/8.0) \* z0;

y1 = y + (126.0/705.0) \* x0 + (9.0/141.0) \* z0;

z1 = z + (3.0/4.0) \* x0 - (1.0/8.0) \* y0;

modx = fabs(x1 - x0);

mody = fabs(y1 - y0);

modz = fabs(z1 - z0);

cout << "|" << setw(2) << k << setw(2)

<< "|" << setw(8) << x0 << setw(5)

<< "|" << setw(10) << x1 << setw(5)

<< "|" << setw(14) << modx << setw(8)

<< "|" << setw(12) << y0 << setw(4)

<< "|" << setw(12) << y1 << setw(4)

<< "|" << setw(14) << mody << setw(7)

<< "|" << setw(12) << z0 << setw(4)

<< "|" << setw(12) << z1 << setw(4)

<< "|" << setw(14) << modz << setw(7)

<< "|" << endl;

k++;

} while ((modx > eps) || (mody > eps) || (modz > eps));

}

void method\_Zeidela() // Метод Зейделя

{

int k = 0;

float x0 = x, x1 = x; // x0 = x(k), x1 = x(k+1)

float y0 = y, y1 = y; // y0 = y(k), y1 = y(k+1)

float z0 = z, z1 = z; // z0 = y(k), z1 = z(k+1)

float modx, mody, modz; // modx = |x1 - x0|, mody = |y1 - y0|, modz = |z1 - z0|

do

{

x0 = x1; y0 = y1; z0 = z1;

x1 = x - 0.1 \* y0 - (1.0 / 8.0) \* z0;

y1 = y + (126.0 / 705.0) \* x1 + (9.0 / 141.0) \* z0;

z1 = z + (3.0 / 4.0) \* x1 - (1.0 / 8.0) \* y1;

modx = fabs(x1 - x0);

mody = fabs(y1 - y0);

modz = fabs(z1 - z0);

cout << "|" << setw(2) << k << setw(2)

<< "|" << setw(8) << x0 << setw(5)

<< "|" << setw(10) << x1 << setw(5)

<< "|" << setw(14) << modx << setw(8)

<< "|" << setw(12) << y0 << setw(4)

<< "|" << setw(12) << y1 << setw(4)

<< "|" << setw(14) << mody << setw(7)

<< "|" << setw(12) << z0 << setw(4)

<< "|" << setw(12) << z1 << setw(4)

<< "|" << setw(14) << modz << setw(7)

<< "|" << endl;

k++;

} while ((modx > eps) || (mody > eps) || (modz > eps));

}

void show\_table()

{

cout << "-----------------------------------------------------------------------------------------------------------------------------------------------------------------\n";

cout << "| k | X(k) | X(k + 1) | |(X(k+1) - X(k))| | Y(k) | Y(k + 1) | |Y(k + 1) - Y(k)| | Z(k) | Z(k + 1) | |Z(k + 1) - Z(k)| |\n";

cout << "-----------------------------------------------------------------------------------------------------------------------------------------------------------------\n";

}

void func()

{

int n;

while (true)

{

cout << "Команда 1 - поиск корня на отрезке по Методу простых итераций" << endl;

cout << "Команда 2 - поиск корня на отрезке по Методу Зейделя" << endl;

cout << "Введите команду: "; cin >> n;

switch (n)

{

case 1:

cout << "\nМетод простых итераций" << endl;

show\_table();

method\_simple\_implications();

cout << "\nПроцесс поиска корня на отрезке прошел успешно!" << endl;

break;

case 2:

cout << "\nМетод Зейделя" << endl;

show\_table();

method\_Zeidela();

cout << "\nПроцесс поиска корня на отрезке прошел успешно!" << endl;

break;

default:

cout << "\nОшибка ввода! Команды не существует! Повторите ввод!\n";

break;

}

cout << "\n";

}

}

int main()

{

setlocale(0, "");

cout << "Лабораторная работа по Вычислительной математике №5\n" << endl;

cout << "Выполнили студенты Маннанов и Нигамадянов гр. 4310\n" << endl;

cout << "Вариант №19\nСЛАУ:\n 5x + 0.5y + 5/8 z = 5;\n 2.8x - 141/9 y + z = 2.8;\n 6x - y - 8z = 6;\n" << endl;

func();

system("pause");

return 0;

}

Работа программы:

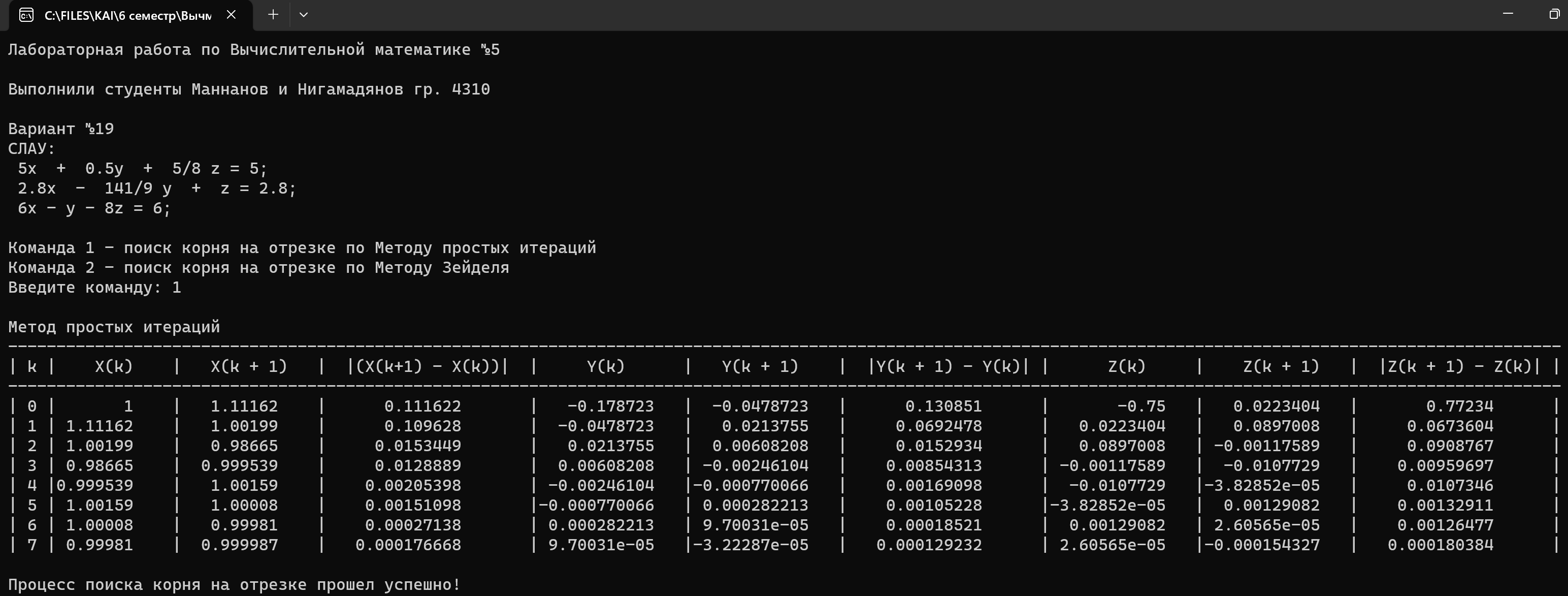


Рис.2

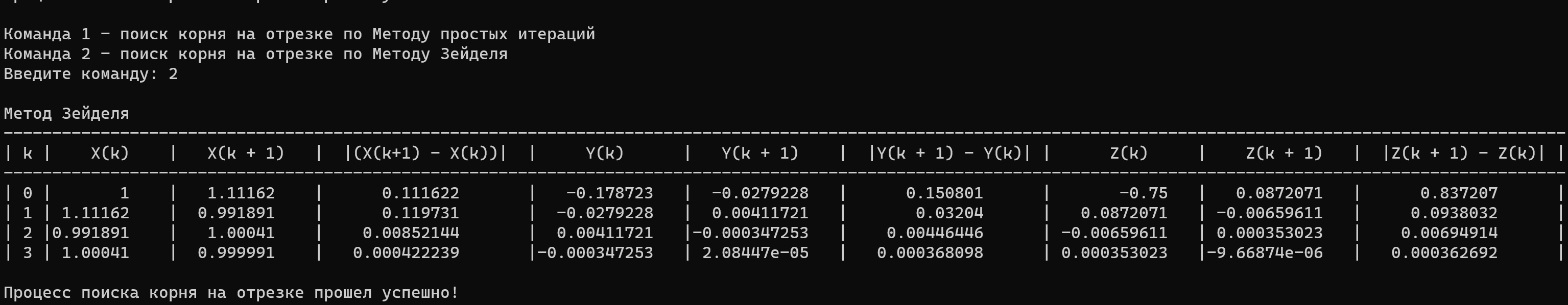


Рис.3

Таблица результатов вычислений по двум методам:

**Метод простых итераций**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
| 0 | 1 | 1.11162 | 0.111622 | -0.178723 | -0.0478723 | 0.130851 | -0.75 | 0.0223404 | 0.77234 |
| 1 | 1.11162 | 1.00199 | 0.109628 | -0.0478723 | 0.0213755 | 0.0692478 | 0.0223404 | 0.0897008 | 0.0673604 |
| 2 | 1.00199 | 0.98655 | 0.0153449 | 0.0213755 | 0.00608208 | 0.0152934 | 0.0897008 | -0.00117589 | 0.0908767 |
| 3 | 0.98655 | 0.999539 | 0.0128889 | 0.00608208 | -0.00246104 | 0.00854313 | -0.00117589 | -0.0107729 | 0.00959697 |
| 4 | 0.999539 | 1.00159 | 0.00205398 | -0.00246104 | -0.000770066 | 0.00169098 | -0.0107729 | -0.0000382852 | 0.0107346 |
| 5 | 1.00159 | 1.00008 | 0.00151098 | -0.000770066 | 0.000282213 | 0.00105228 | -0.0000382852 | 0.00129082 | 0.00132911 |
| 6 | 1.00008 | 0.99981 | 0.00027138 | 0.000282213 | 0.0000970031 | 0.00018521 | 0.00129082 | 0.0000260565 | 0.00126477 |
| 7 | 0.99981 | 0.999987 | 0.000176668 | 0.0000970031 | -0.0000322287 | 0.000129232 | 0.0000260565 | -0.000154327 | 0.000180384 |

**Метод Зейделя**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
| 0 | 1 | 1.11162 | 0.111622 | -0.178723 | -0.0279228 | 0.150801 | -0.75 | 0.0872071 | 0.837207 |
| 1 | 1.11162 | 0.991891 | 0.119731 | -0.0279228 | 0.00411721 | 0.03204 | 0.0872071 | -0.00659611 | 0.0938032 |
| 2 | 0.991891 | 1.00041 | 0.00852144 | 0.00411721 | -0.000347253 | 0.00446446 | -0.00659611 | 0.000353023 | 0.00694914 |
| 3 | 1.00041 | 0.999991 | 0.000422239 | -0.000347253 | 0.0000208447 | 0.000368098 | 0.000353023 | 0.00000966874 | 0.000362692 |

Вывод: в результате решения СЛАУ (1) методом простых итераций с точностью получено решение , методом Зейделя с той же точностью .