Группа 4310

Нигамадянов Ф.М.

Маннанов Д.И.

**ЛАБОРАТОРНАЯ РАБОТА №3-4.**

**ИТЕРАЦИОННЫЕ МЕТОДЫ РЕШЕНИЯ СИСТЕМ НЕЛИНЕЙНЫХ УРАВНЕНИЙ**

Цель работы: научиться решать системы нелинейных уравнений (СНУ) методом простых итераций (МПИ) и методом Ньютона с помощью ЭВМ.

Содержание работы:

1. Изучить МПИ и метод Ньютона для решения систем нелинейных уравнений.
2. На конкретном примере усвоить порядок решения систем нелинейных уравнений МПИ и методом Ньютона с помощью ЭВМ.
3. Составить программу и с ее помощью решить систему уравнений с точностью .
4. Составить отчет о проделанной работе.

Задание (вариант №20).

1. Аналитически решить СНУ вида:

(1)

2. Построить рабочие формулы МПИ и метода Ньютона для численного решения системы (1) при начальном приближении

(2)

3. Составить программу на любом языке программирования, реализующую построенный итерационный процесс.

Решение.

1. Аналитическим решением СНУ (1) являются точки и .

2. Для построения рабочих формул МПИ для численного решения системы (1) необходимо вначале привести ее к виду:

(3)

Для этого умножим первое уравнение системы (1) на неизвестную постоянную α , второе – на β, затем сложим их и добавим в обе части уравнения . Получим первое уравнение преобразуемой системы:

(4)

где . Далее, умножим первое уравнение системы (1) на неизвестную постоянную γ, второе - на δ, затем сложим их и добавим в обе части уравнения . Тогда второе уравнение преобразуемой системы будет иметь вид:

(5)

где .

Неизвестные постоянные α, β, γ, δ определим из достаточных условий сходимости итерационного процесса:

.

Запишем эти условия более подробно:

Полагая равными нулю выражения под знаком модуля, получим систему линейных алгебраических уравнений (СЛАУ) 4 порядка с 4 неизвестными α, β, γ, δ:

(6)

Для решения системы (6) необходимо вычислить частные производные :

Тогда СЛАУ (6) запишется так:

Заметим, что частные производные мало изменяются в окрестности начального приближения (2). В таком случае:

Тогда СЛАУ (6) запишется так:

Решением этой системы являются точки

. Тогда рабочие формулы (4), (5) МПИ для решения СНУ (1) примут вид:

Для реализации на ЭВМ рабочие формулы можно переписать так:

(7)

Итерационный процесс (7) можно начать, задав начальное приближение (2). Процесс (7) заканчивается при одновременном выполнении двух условий: и . В этом случае значения и являются приближенным значением одного из решений СНУ (1).

3. Для построения рабочих формул метода Ньютона в виде

(8)

причем

и

необходимо:

1. Найти матрицу частных производных:

2. Найти определитель этой матрицы:

3. Определить обратную матрицу:

Проведя несложные преобразования получим рабочую формулу метода Ньютона (8) в виде:

(9)

Или

(10)

Листинг программы:

#include <iostream>

#include "locale.h"

#include <math.h>

#include <iomanip>

using namespace std;

const float eps = 0.001; // Точность

const float x = 1.8, y = 1; // Начальное приближение

// коэффициенты для МПИ:

const float alpha = -5.0/18;

const float beta = 0.0;

const float gamma = 5.0/48;

const float delta = -1.0/3;

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 modx, mody; // modx: |x1 - x0|, mody: |y1 - y0|

do

{

x0 = x1, y0 = y1;

x1 = x0 + alpha \* (pow(x0, 2) - 4.0) + beta \* (x0 + 3 \* y0 - 8.0);

y1 = y0 + gamma \* (pow(x0, 2) - 4.0) + delta \* (x0 + 3 \* y0 - 8.0);

modx = fabs(x1 - x0);

mody = fabs(y1 - y0);

cout << "|" << setw(4) << k << setw(4) << "|" << setw(8) << x0 << setw(5) << "|" << setw(10) << x1 << setw(5) << "|"

<< setw(14) << modx << setw(8) << "|" << setw(9) << y0 << setw(4) << "|" << setw(9) << y1 <<

setw(4) << "|" << setw(11) << mody << setw(7) << "|" << endl;

k++;

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

}

void method\_Newton() // Метод Ньютона

{

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

do // modx: |x1 - x0|, mody: |y1 - y0|

{

x0 = x1, y0 = y1;

x1 = x0 - (3.0 \* (pow(x0, 2) - 4.0) + 0.0 \* (x0 + 3 \* y0 - 8.0)) / (6 \* x0);

y1 = y0 - ((-1.0) \* (pow(x0, 2) - 4.0) + (2.0 \* x0) \* (x0 + 3 \* y0 - 8.0)) / (6 \* x0);

modx = fabs(x1 - x0);

mody = fabs(y1 - y0);

cout << "|" << setw(4) << k << setw(4) << "|" << setw(8) << x0 << setw(5) << "|" << setw(10) <<

x1 << setw(5) << "|" << setw(14) << modx << setw(8) << "|" << setw(9) <<

y0 << setw(4) << "|" << setw(9) << y1 << setw(4) << "|" << setw(11) <<

mody << setw(7) << "|" << endl;

k++;

} while ((modx > eps) || (mody > 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)| |\n";

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

}

void func()

{

int n;

while (true)

{

cout << "Команда 1 - поиск корней на отрезке по МПИ" << endl;

cout << "Команда 2 - поиск корней на отрезке по Методу Ньютона" << 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 Ньютона" << endl;

show\_table();

method\_Newton();

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

break;

default:

cout << "Ошибка ввода !";

break;

}

cout << "\n";

}

}

int main()

{

setlocale(LC\_ALL, "ru");

cout << "Лабораторная работа по Вычислительной математике №3-4\n" << endl;

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

cout << "Вариант №20\nСистема нелинейных уравнений:" << endl;

cout << "x^2 - 4 = 0\nx + 3y - 8 = 0\nНачальное приближение: ( 1.8; 1 )\n" << endl;

func();

system("pause");

return 0;

}

Работа программы:

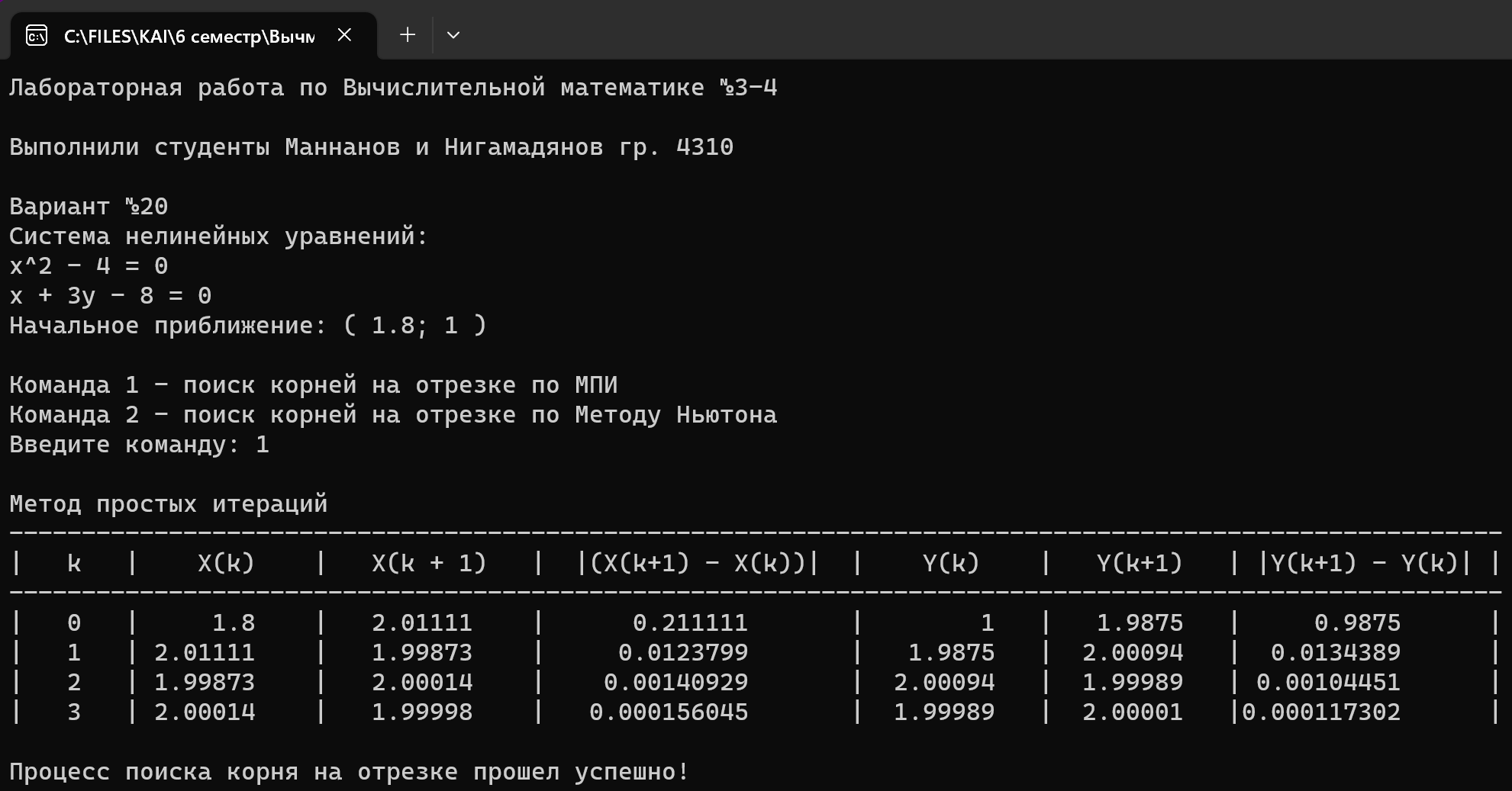


Рис.1

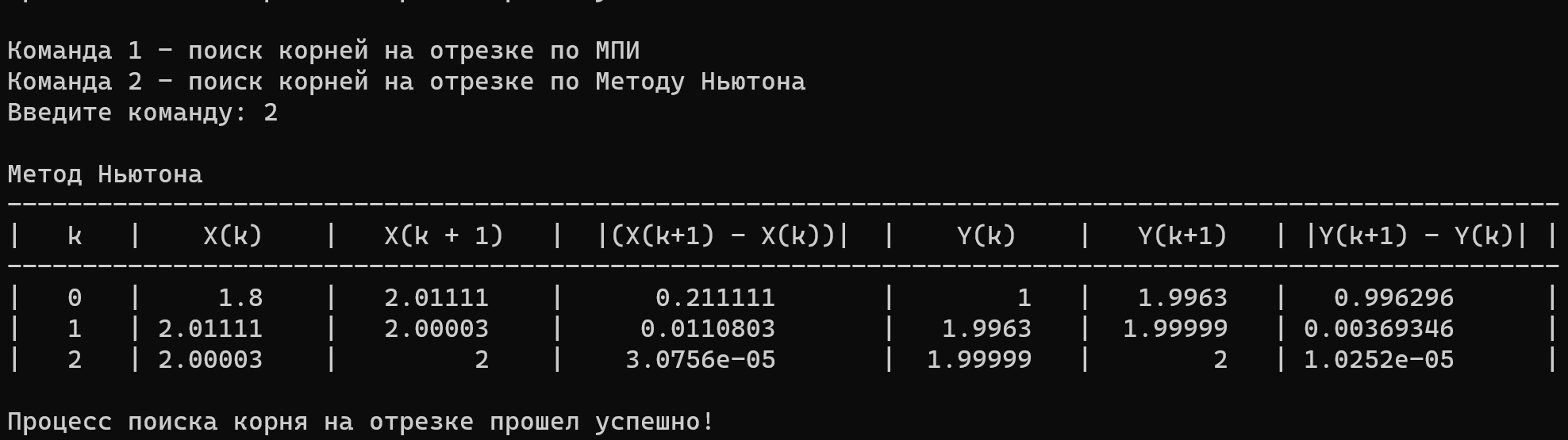


Рис.2

Таблица результатов вычислений по двум методам:

**Метод простых итераций**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| 0 | 1.8 | 2.01111 | 0.211111 | 1 | 1.9875 | 0.9875 |
| 1 | 2.01111 | 1.99873 | 0.0123799 | 1.9875 | 2.00094 | 0.0134389 |
| 2 | 1.99873 | 2.00014 | 0.00140929 | 2.00094 | 1.99989 | 0.00104451 |
| 3 | 2.00014 | 1.99998 | 0/000156045 | 1.99989 | 2.00001 | 0.000117302 |

**Метод Ньютона**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| 0 | 1.8 | 2.01111 | 0.211111 | 1 | 1.9963 | 0.996296 |
| 1 | 2.01111 | 2.00003 | 0.0110803 | 1.9963 | 1.99999 | 0.00369346 |
| 2 | 2.00003 | 2 | 0.000030756 | 1.99999 | 2 | 0.000010252 |

Вывод: в результате решения СНУ (1) при начальном приближении (2) методом простых итераций с точностью получено решение , и методом Ньютона .