**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ**

Лабораторная №2

**«Численное решение систем нелинейных уравнений»**

**Выполнил:**

Крученков Евгений Андреевич

студент 2 курса 9 группы,

специальность

“прикладная математика”

**Преподаватель:**

Ассистент кафедры вычислительной

математики ФПМИ,

Ю.Н. Горбачёва

Минск, 2022

**Содержание:**

Постановка задачи ------------------------------------------------------------------ 2

Краткие теоретические сведения ------------------------------------------------ 3-4

Листинг программы ---------------------------------------------------------------- 5-11

Результаты --------------------------------------------------------------------------- 11-12

Выводы ------------------------------------------------------------------------------- 13

**Постановка задачи**

**Задание 1**

Написать программу, которая решает данную систему нелинейных уравнений f(x) = 0 c точностью ε = с помощью метода Ньютона, метода секущих. Начальное приближение выбрать графически. Провести сравнительный анализ полученных результатов.

В содержание отчета должна быть включена следующая информация:

• График для нахождения начального приближения.

• Алгоритм метода Ньютона.

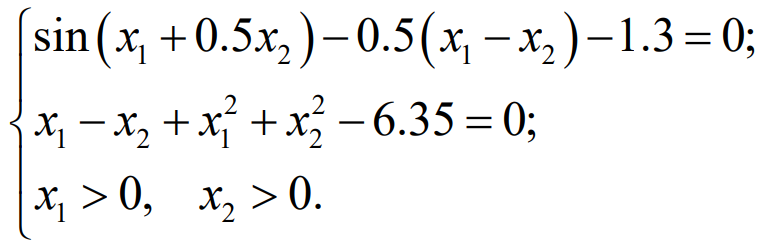
• Алгоритм метода секущих.

• Результаты вычислительного эксперимента в виде таблицы.

• Для каждого метода указать значение где - полученное решение.

• Листинг программы с комментариями.

На 9–10 баллов необходимо также решить систему методом Гаусса-Зейделя (внутренний итерационный процесс – метод Ньютона). В содержание отчета включить алгоритм метода Гаусса-Зейделя и полученные результаты.

****

**Краткие теоретические сведения**

**МЕТОД НЬЮТОНА**

J = - матрица Якоби

= +

K = 0,1,…

Условие остановки:

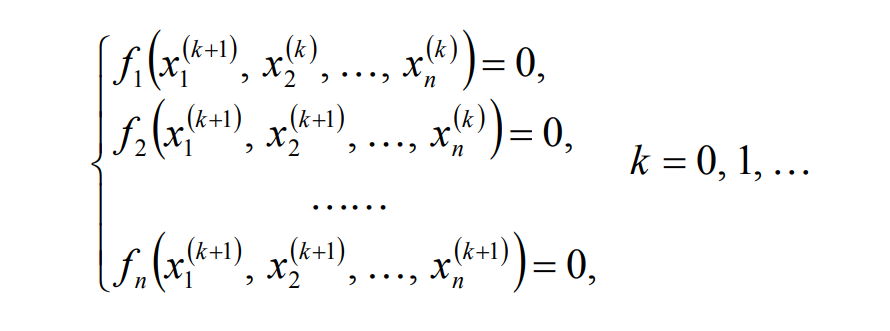
**МЕТОД СЕКУЩИХ**

J = - матрица Якоби

K = 1,2,…

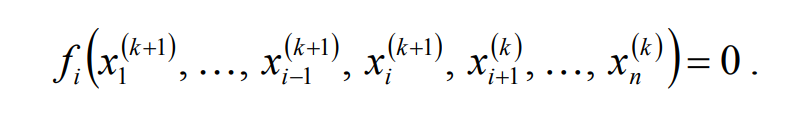
Процесс останавливается, когда

**МЕТОД ГАУССА-ЗЕЙДЕЛЯ**

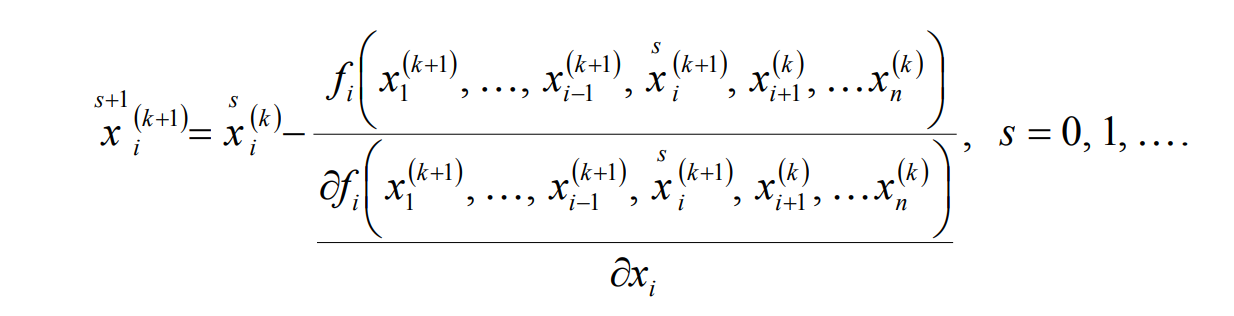
****

Внешний цикл:

Для нахождения на каждом шаге, требуется решить нелинейное уравнение с одной неизвестной.



Внутренний цикл по методу Ньютона:



Условие остановки внутреннего цикла:

Условие остановки внешнего цикла:

**Листинг программы**

**Header.h**

#pragma once

#include <iostream>

#include <stdio.h>

#include <iomanip>

#include <fstream>

#include <string>

#include <time.h>

using namespace std;

double E = pow(10, -6);

// Гаусс

class Matr

{

friend class Sist;

double\*\* M;

int n;

public:

Matr()

{

this->n = 1;

M = new double\* [n];

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

{

M[i] = new double[n];

}

srand(time(0));

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

{

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

{

M[i][j] = (((double)rand() / RAND\_MAX) \* 2 - 1) \* 100;

}

}

}

Matr(int n, double\*\* M1)

{

this->n = n;

M = new double\* [n];

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

{

M[i] = new double[n];

}

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

{

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

{

M[i][j] = M1[i][j];

}

}

}

Matr(const Matr& B)

{

this->n = B.n;

M = new double\* [n];

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

{

M[i] = new double[n];

}

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

{

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

{

M[i][j] = B.M[i][j];

}

}

}

~Matr()

{

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

{

delete[] M[i];

}

delete[] M;

}

friend ostream& operator <<(ostream& os, Matr& A)

{

os << fixed << showpoint << setprecision(3);

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

{

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

{

os << A.M[i][j] << " ";

}

os << endl;

}

return os;

}

Matr& operator = (const Matr& B)

{

if (this == &B)

return \*this;

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

{

delete[] M[i];

}

n = B.n;

delete[] M;

M = new double\* [n];

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

{

M[i] = new double[n];

}

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

{

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

{

M[i][j] = B.M[i][j];

}

}

return \*this;

}

};

class Sist

{

friend class Matr;

private:

Matr A;

double\* X;

double\* F;

double\* X\_;

double pogr;

public:

int size()

{

return A.n;

}

void ShowPogr()

{

cout << scientific << pogr << fixed << showpoint << setprecision(3);

}

Sist(const Matr& B, double\* F1)

{

A = B;

X = new double[B.n];

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

{

X[i] = 1 + i;

}

F = F1;

X\_ = new double[B.n];

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

{

X\_[i] = 0;

}

}

~Sist()

{

delete[] X;

delete[] X\_;

delete[] F;

}

double\* Gauss()

{

//выбор главного элемента по столбцу

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

{

double max = abs(A.M[i][i]);

int gl = i;

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

{

if (abs(A.M[j][i]) > max)

{

max = A.M[j][i];

gl = j;

}

}

if (max == 0) throw "Все главные элементы нулевые";

//исключение в случае, когда все главные элементы нулевые

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

{

swap(A.M[i][j], A.M[gl][j]);

}

swap(F[i], F[gl]);

//прямой ход метода Гаусса

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

{

A.M[i][j] /= A.M[i][i];

}

F[i] /= A.M[i][i];

A.M[i][i] /= A.M[i][i];

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

{

for (int k = i + 1; k < size(); k++)

{

A.M[j][k] -= A.M[i][k] \* A.M[j][i];

}

F[j] -= F[i] \* A.M[j][i];

A.M[j][i] -= A.M[i][i] \* A.M[j][i];

}

}

//обратный ход метода Гаусса

for (int i = (size() - 1); i >= 0; i--)

{

X\_[i] = F[i];

for (int j = size() - 1; j > i; j--)

{

X\_[i] -= A.M[i][j] \* X\_[j];

}

}

//вычисление относительной погрешности

double maxX = abs(X[0]), max\_X = abs(X[0] - X\_[0]);

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

{

if (max\_X < abs(X[i] - X\_[i]))

{

max\_X = abs(X[i] - X\_[i]);

}

if (maxX < abs(X[i]))

{

maxX = abs(X[i]);

}

}

pogr = max\_X / maxX;

return X\_;

}

};

double F1(double x, double y) {

return (sin(x + 0.5 \* y) - 0.5\*(x - y) -1.3);

}

double F2(double x, double y) {

return (x-y+x\*x+y\*y-6.35);

}

double DF1\_Dx(double x, double y) {

return (cos(x + 0.5 \* y) - 0.5);

}

double DF1\_Dy(double x, double y) {

return (0.5\*cos(x + 0.5 \* y) + 0.5);

}

double DF2\_Dx(double x, double y) {

return (1+2\*x);

}

double DF2\_Dy(double x, double y) {

return (2\*y-1);

}

void Newton(double x\_0, double y\_0) {

double x\_k, y\_k, x\_k\_1, y\_k\_1, delta\_x\_k, delta\_y\_k;

x\_k = x\_0;

y\_k = y\_0;

cout <<"k"<<" " << " x\_k" << " " << "y\_k" << " " << " || delta\_X || " << endl;

cout <<"0 " << x\_0 << " " << y\_0 << " " << "-" << endl;

int k = 1;

while (true) {

double\*\* M = new double\*[2];

M[0] = new double[2];

M[1] = new double[2];

M[0][0] = DF1\_Dx(x\_k, y\_k);

M[0][1] = DF1\_Dy(x\_k, y\_k);

M[1][0] = DF2\_Dx(x\_k, y\_k);

M[1][1] = DF2\_Dy(x\_k, y\_k);

double\* F = new double[2];

F[0] = (-1) \* F1(x\_k, y\_k);

F[1] = (-1) \* F2(x\_k, y\_k);

Matr A(2, M);

Sist S(A, F);

double\* delta\_X = S.Gauss();

x\_k\_1 = x\_k + delta\_X[0];

y\_k\_1 = y\_k + delta\_X[1];

double norm = abs(delta\_X[0]);

if (abs(delta\_X[1]) > norm) norm = abs(delta\_X[1]);

cout <<dec<<k<<" "<<scientific<< x\_k\_1 << " " << y\_k\_1 << " " << norm <<dec<< endl;

x\_k = x\_k\_1;

y\_k = y\_k\_1;

k++;

if (norm <= E) break;

}

double norm\_f = F1(x\_k\_1, y\_k\_1);

if(F2(x\_k\_1, y\_k\_1)> norm\_f) norm\_f = F2(x\_k\_1, y\_k\_1);

cout << "||F(X\_n)|| = " << norm\_f<<endl;

}

void Sek(double x\_0, double y\_0, double x\_1, double y\_1) {

double x\_k, y\_k, x\_k\_1, y\_k\_1, x\_k\_2, y\_k\_2, delta\_x\_k, delta\_y\_k;

x\_k = x\_0;

y\_k = y\_0;

x\_k\_1 = x\_1;

y\_k\_1 = y\_1;

cout <<dec<< "k" << " " << " x\_k" << " " << "y\_k" << " " << " || delta\_X || " << endl;

cout << "0 " << x\_0 << " " << y\_0 << " " << "-" << endl;

double norm1 = abs(x\_1 - x\_0);

if (abs(y\_1 - y\_0) > norm1) norm1 = abs(y\_1 - y\_0);

cout << "1 " << x\_1 << " " << y\_1 << " " << norm1 << endl;

int k = 2;

while (true) {

double\*\* M = new double\* [2];

M[0] = new double[2];

M[1] = new double[2];

M[0][0] = (F1(x\_k\_1, y\_k\_1) - F1(x\_k, y\_k\_1)) / (x\_k\_1 - x\_k);

M[0][1] = (F1(x\_k\_1, y\_k\_1) - F1(x\_k\_1, y\_k)) / (y\_k\_1 - y\_k);

M[1][0] = (F2(x\_k\_1, y\_k\_1) - F2(x\_k, y\_k\_1)) / (x\_k\_1 - x\_k);

M[1][1] = (F2(x\_k\_1, y\_k\_1) - F2(x\_k\_1, y\_k)) / (y\_k\_1 - y\_k);

double\* F = new double[2];

F[0] = (-1) \* F1(x\_k\_1, y\_k\_1);

F[1] = (-1) \* F2(x\_k\_1, y\_k\_1);

Matr A(2, M);

Sist S(A, F);

double\* delta\_X = S.Gauss();

x\_k\_2 = x\_k\_1 + delta\_X[0];

y\_k\_2 = y\_k\_1 + delta\_X[1];

double norm = abs(delta\_X[0]);

if (abs(delta\_X[1]) > norm) norm = abs(delta\_X[1]);

cout << dec << k << " " << scientific << x\_k\_2 << " " << y\_k\_2 << " " << norm <<dec<< endl;

x\_k = x\_k\_1;

y\_k = y\_k\_1;

x\_k\_1 = x\_k\_2;

y\_k\_1 = y\_k\_2;

k++;

if (norm <= E) break;

}

double norm\_f = F1(x\_k\_2, y\_k\_2);

if (F2(x\_k\_2, y\_k\_2) > norm\_f) norm\_f = F2(x\_k\_2, y\_k\_2);

cout << "||F(X\_n)|| = " << norm\_f << endl;

}

void Gauss\_Zeidel(double x\_0, double y\_0) {

double x\_k, y\_k, x\_k\_1, y\_k\_1, norm\_f;

x\_k = x\_0;

y\_k = y\_0;

cout << dec << "k" << " " << " x\_k" << " " << "y\_k" << " " << " ||F(X\_k)|| " << endl;

double norm\_f1 = F1(x\_k, y\_k);

if (F2(x\_k, y\_k) > norm\_f1) norm\_f1 = F2(x\_k, y\_k);

cout << "0 " << x\_0 << " " << y\_0 << " " << norm\_f1 << endl;

int k = 1;

while (true) {

//1

double x\_s, x\_s\_1;

x\_s = x\_k;

while (true)

{

x\_s\_1 = x\_s - (F1(x\_s, y\_k) / DF1\_Dx(x\_s, y\_k));

if (abs(x\_s - x\_s\_1) < E) break;

x\_s = x\_s\_1;

}

x\_k\_1 = x\_s\_1;

//

//

//2

double y\_s, y\_s\_1;

y\_s = y\_k;

while (true)

{

y\_s\_1 = y\_s - (F2(x\_k\_1, y\_s) / DF2\_Dy(x\_k\_1, y\_s));

if (abs(y\_s - y\_s\_1) < E) break;

y\_s = y\_s\_1;

}

y\_k\_1 = y\_s\_1;

//

x\_k = x\_k\_1;

y\_k = y\_k\_1;

norm\_f = abs(F1(x\_k\_1, y\_k\_1));

if (abs(F2(x\_k\_1, y\_k\_1)) > norm\_f) norm\_f = abs(F2(x\_k\_1, y\_k\_1));

cout << dec << k << " " << scientific << x\_k\_1 << " " << y\_k\_1 << " " << norm\_f << dec << endl;

k++;

if (norm\_f <= E) break;

}

}

**Main.cpp**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include "Header.h"

#include <fstream>

#include<time.h>

using namespace std;

int main()

{

setlocale(LC\_ALL, "ru");

cout << "Метод Ньютона:" << endl;

Newton(1, 2);

cout << endl << "Метод секущих:" << endl;

Sek(1, 2, 1.2, 2.4);

cout << endl << "Метод Гаусса-Зейделя:" << endl;

Gauss\_Zeidel(1, 2);

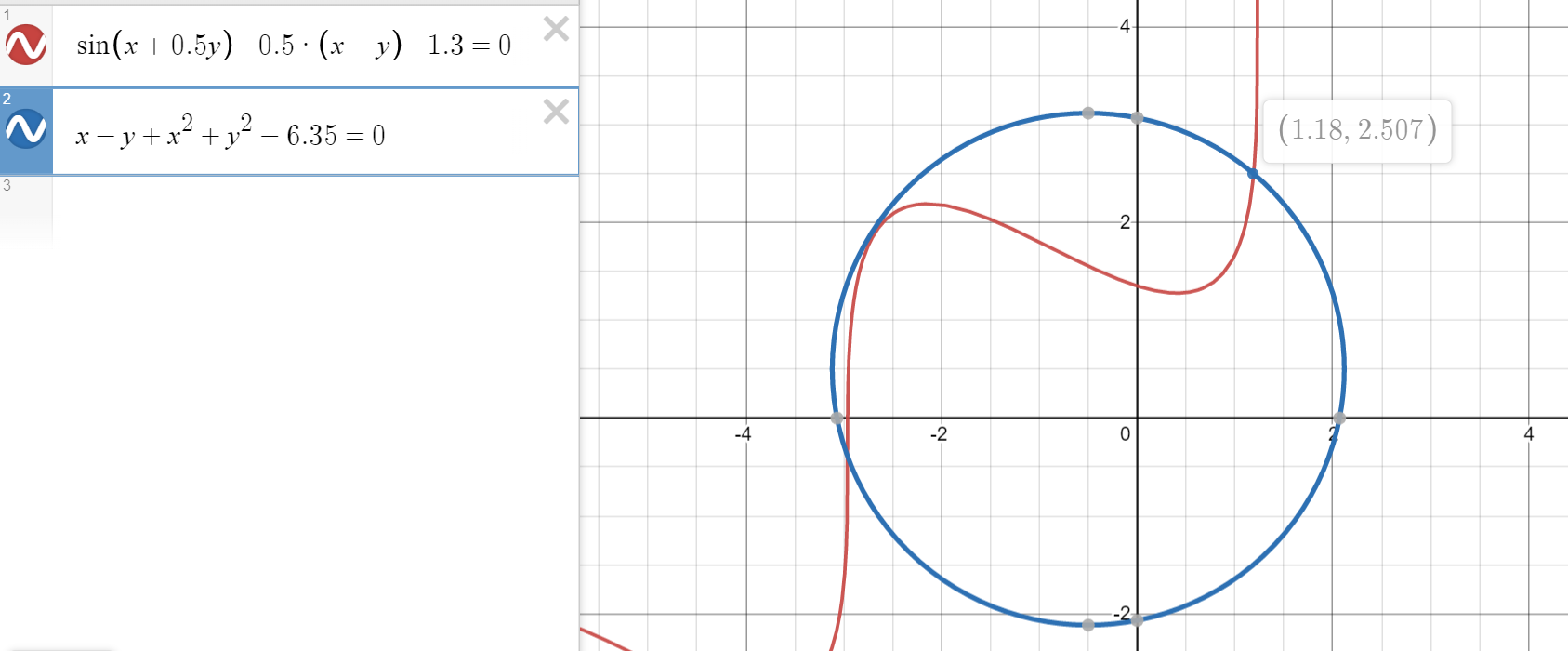
system("pause");

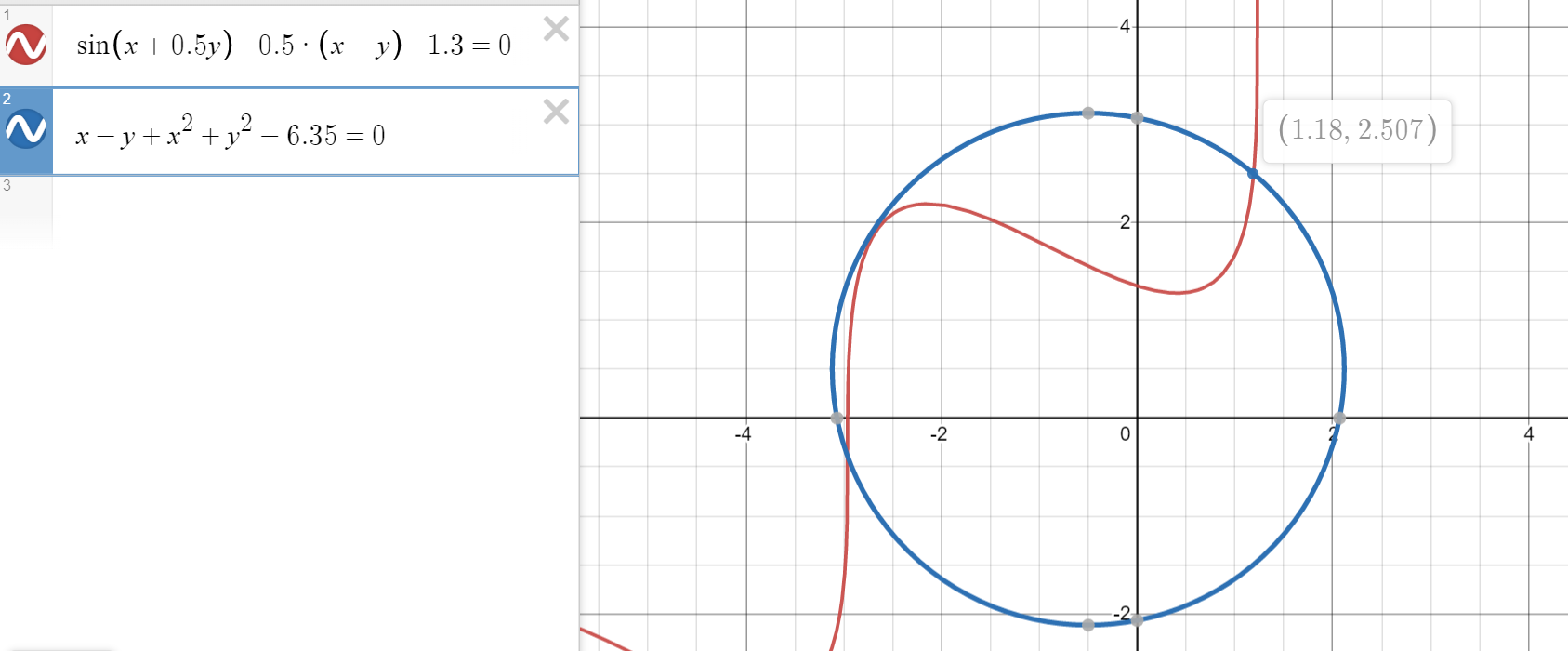
return 0;

}

**Результаты**

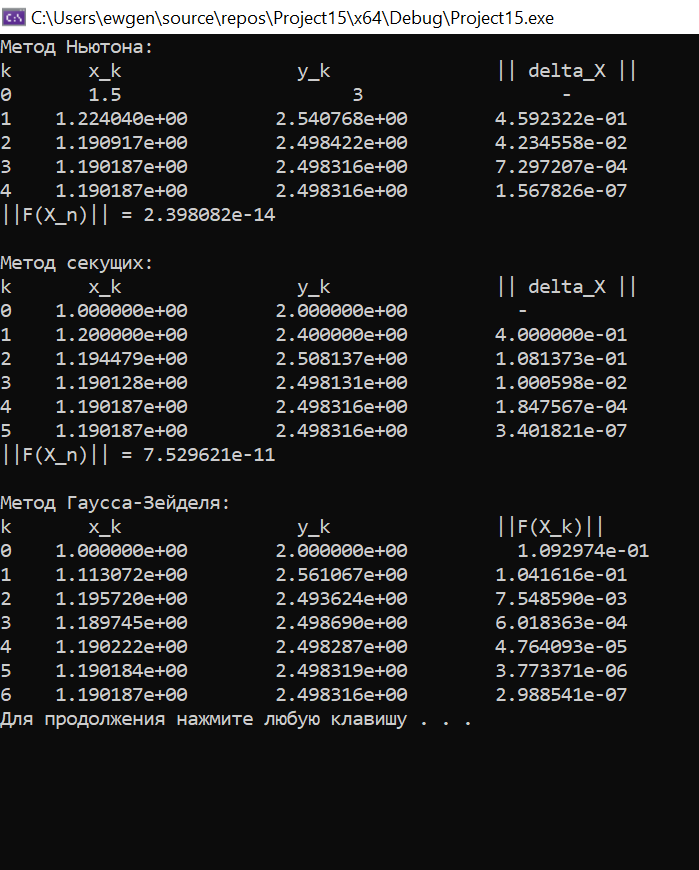
*График функции:*

****

****

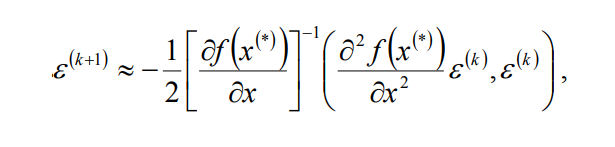
Выберем начальное приближение = (1, 2).

Второе приближение для метода секущих = (1.2, 2.4).

****

**Выводы**

1. Погрешность метода Ньютона на k+1 шаге выражается через погрешность на k шаге следующим образом:



Следовательно, метод имеет квадратичную сходимость.

1. В отличие от метода Зейделя, система в методе Гаусса-Зейделя не требует предварительного преобразования системы к каноническому виду, что весьма удобно.