|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана (национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

|  |  |
| --- | --- |
| ФАКУЛЬТЕТ | *Робототехники и комплексной автоматизации* |
|  |  |
| КАФЕДРА | *Системы автоматизированного проектирования (РК-6)* |

**ТЕХНОЛОГИЧЕСКАЯ ПРАКТИКА ПО ТЕМЕ:**

**ПАРАМЕТРИЧЕСКАЯ ОПТИМИЗАЦИЯ И РЕАЛИЗАЦИЯ ЕЕ МЕТОДОВ**

|  |  |  |
| --- | --- | --- |
| Студент |  | Жарова Татьяна Александровна |
| Группа |  | РК6-45Б |

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Жарова Т.А.**

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Волосатова Т.М.**

*подпись, дата фамилия, и.о.*

Консультант **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Дорофеев В.С.**

*подпись, дата фамилия, и.о.*

*Москва, 2022 г.*

**СОДЕРЖАНИЕ:**

[ЦЕЛИ И ЗАДАЧИ: 3](#_Toc104995534)

[ТЕХНИЧЕСКОЕ ЗАДАНИЕ: 3](#_Toc104995535)

[1. ОБЩЕЕ ОПИСАНИЕ РАБОТЫ: 3](#_Toc104995536)

[1.1. ПРОГРАММНОЕ И ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ 3](#_Toc104995537)

[1.2. ИСПОЛЬЗОВАНИЕ БИБЛИОТЕК 4](#_Toc104995538)

[1.3. ТЕСТОВЫЕ ФУНКЦИИ 4](#_Toc104995539)

[1.4. ЭКСПЕРИМЕНТАЛЬНАЯ ПРОВЕРКА РАБОТЫ 7](#_Toc104995540)

[1.5. СРАВНЕНИЕ РЕЗУЛЬТАТОВ 7](#_Toc104995541)

[ВЫВОД: 7](#_Toc104995542)

[ПРИЛОЖЕНИЕ А: 8](#_Toc104995543)

[ЛИСТИНГ ПРОГРАММЫ ЗОЛОТОГО СЕЧЕНИЯ: 8](#_Toc104995544)

[ЛИСТИНГ ПРОГРАММЫ СЕТОЧНОГО ПОИСКА: 9](#_Toc104995545)

[ПРИЛОЖЕНИЕ Б: 10](#_Toc104995546)

[РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ПРОГРАММЫ ЗОЛОТОГО СЕЧЕНИЯ: 10](#_Toc104995547)

[РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ПРОГРАММЫ СЕТОЧНОГО ПОИСКА: 10](#_Toc104995548)

# **ЦЕЛИ И ЗАДАЧИ:**

***Цель разработки:*** изучить методы параметрической оптимизации, реализовать программу поиска минимума функции для двух из них, используя нескольких тестовых функций.

Мы будем рассматривать два метода: сеточный поиск и золотое сечение.

***Задачи данной работы:***

* Изучить алгоритмы сеточного поиска и золотого сечения;
* Разработать программу двух методов для нескольких тестовых функций.
* Сравнить точность результатов;
* Составить вывод о результативности данной работы.

# **ТЕХНИЧЕСКОЕ ЗАДАНИЕ:**

Разработать программы для сеточного поиска и золотого сечения, в которых будут рассмотрены 2 функции и для каждой из них будет реализован поиск минимума. Сравнить точность и сделать вывод об эффективности данных методов.

# **ОБЩЕЕ ОПИСАНИЕ РАБОТЫ:**

1. Разработали программу методов золотого сечения и сеточного поиска с разными функциями цели.
2. Сравнили точность выполнения программы.
3. Сделать вывод об эффективности программы.

## **ПРОГРАММНОЕ И ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ**

Для реализации программы использовались: операционная система Ubunta, текстовый редактор и компилятор.

Центральный процессор – Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz

Графический процессор – Intel(R) UHD Graphics 620 (WHL GT2)

Оперативная память – 8 ГБ

## **ИСПОЛЬЗОВАНИЕ БИБЛИОТЕК**

В программах использовались библиотеки языка С++

<iostream> - библиотека для ввода и вывода, а также для тестовых функций

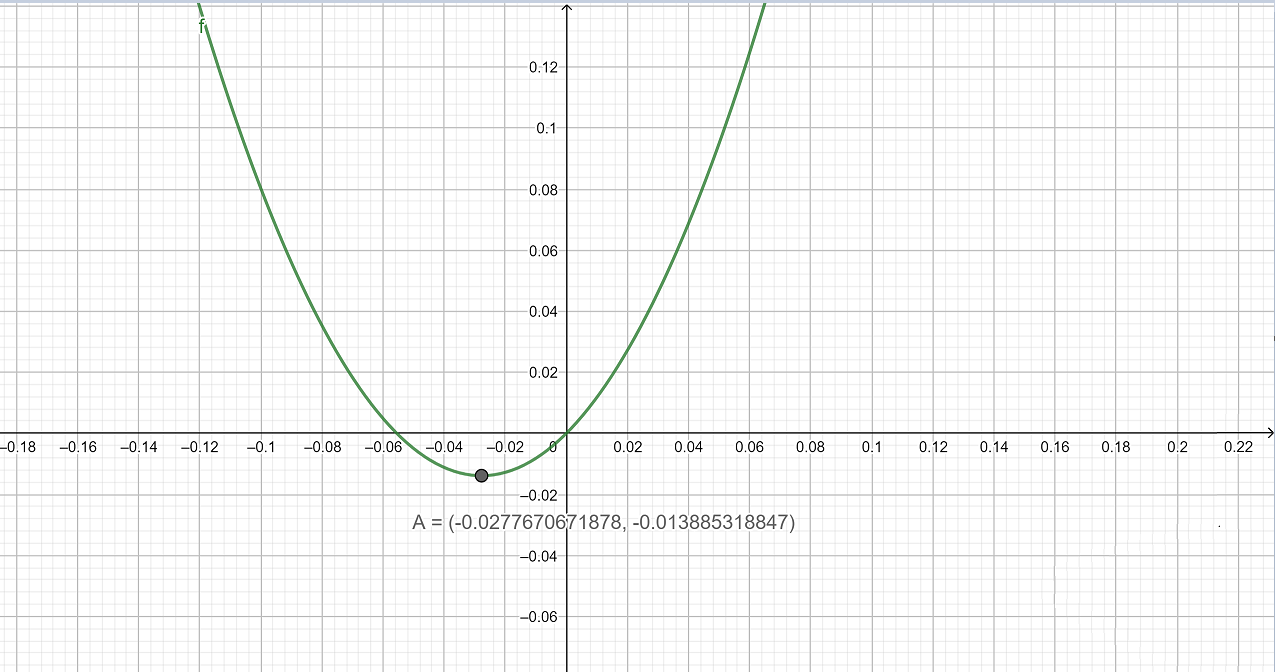
<cmath> - библиотека для вычисления общих математических операций и преобразований:

## **ТЕСТОВЫЕ ФУНКЦИИ**

В данной работе я использовала 2 функции:

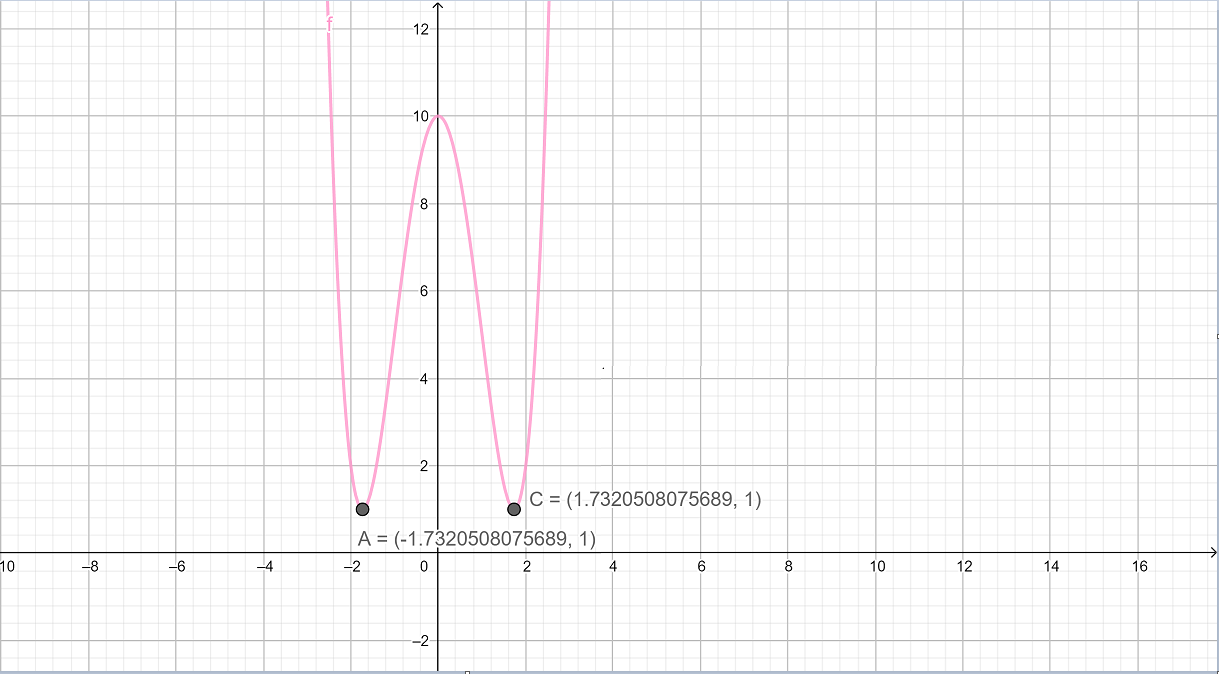
Я искала экстремумы на отрезке [-10;10]

На рис. 1 показан график функции . У нее всего один экстремум на отрезке [-10;10]. Точка А с координатам (-0.0278;-0.0139) является минимумом данной функции



*Рис 1. График функции и ее минимум*

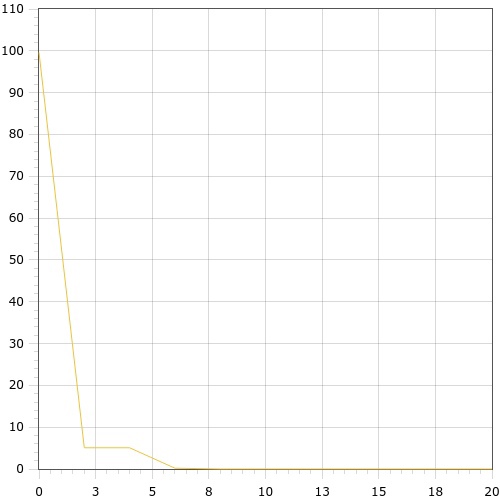
На рис. 2 показан график функции . У нее три экстремума на отрезке [-10;10]. Точки А и C с координатами (-1,7321;1) и (1,7321;1) являются минимумами данной функции.

**

*Рис 2. График функции и ее минимум*

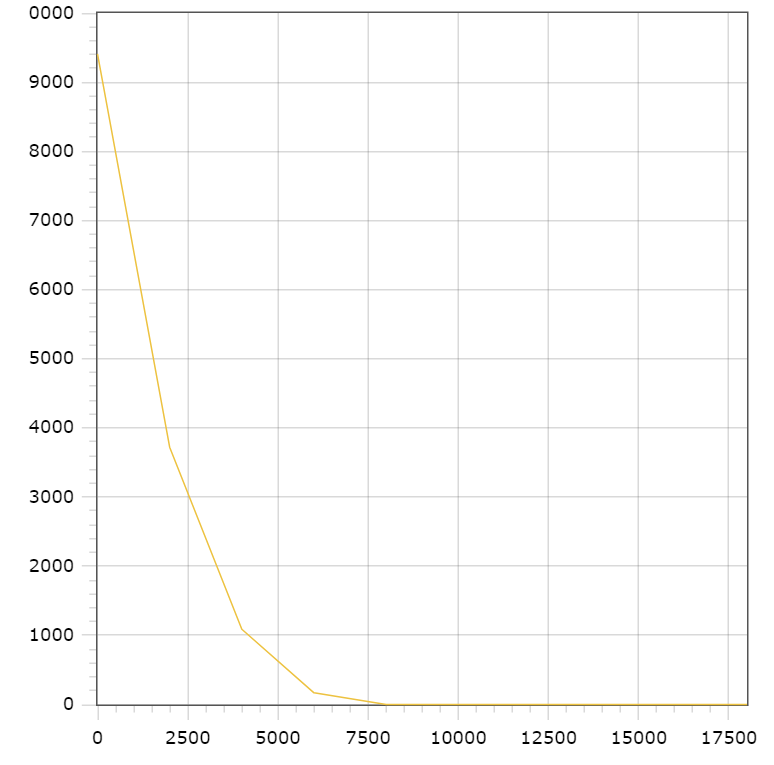
Ниже приведены графики сходимости по итерациям.

Координата x – порядковый номер итерации. Координата y – разность значения функции при данной итерации и настоящего минимума.



*Рис 3.График сходимости функции методом золотого сечения*

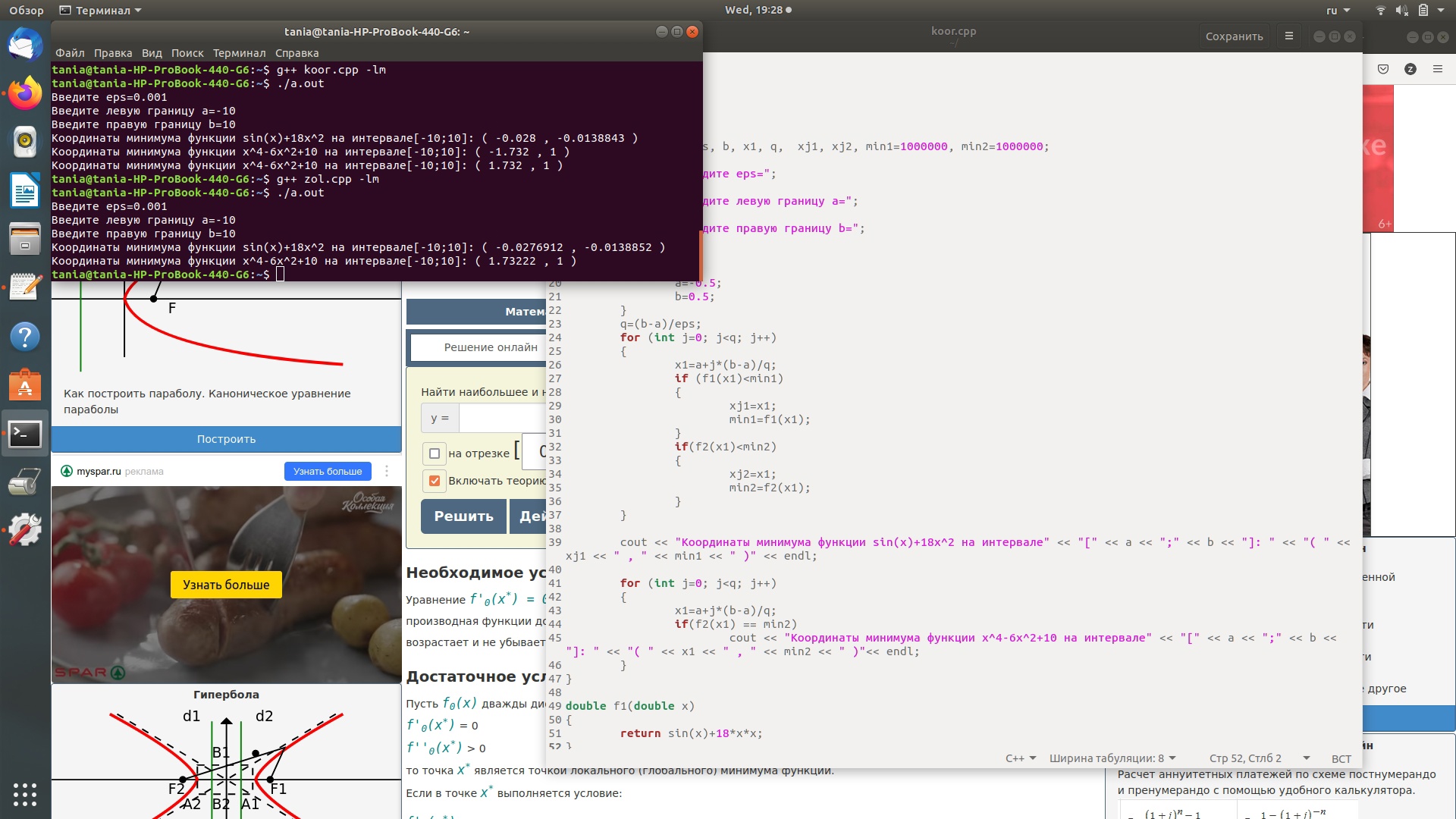
10000



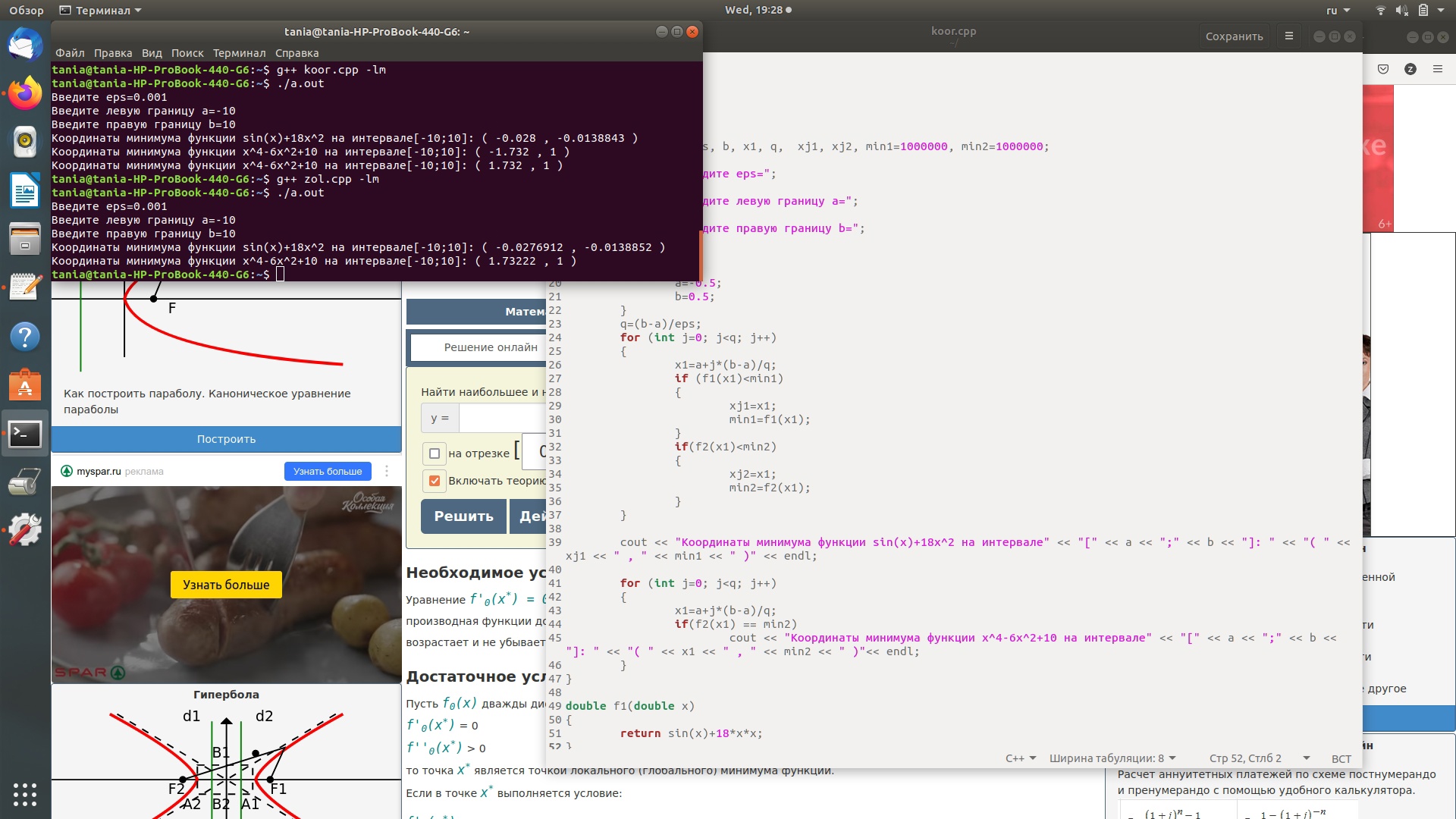
*Рис 4.График сходимости функции методом сеточного поиска*

## **ЭКСПЕРИМЕНТАЛЬНАЯ ПРОВЕРКА РАБОТЫ**

Результат выполнения программы методом золотого сечения:



Результат выполнения программы методом сеточного поиска:



## **СРАВНЕНИЕ РЕЗУЛЬТАТОВ**

Как мы можем заметить результаты обоих методов близки к реальным значениям, но более точное решение представлено в золотом сечении. Сеточному поиску же не требуется предварительно выделять унимодальные области оптимизируемой функции, поэтому он нашел все минимумы . Как было сказано ранее эффективнее является метод золотое сечение, так как он тратит меньше времени при большем количестве комбинаций. Сеточный поиск же обладает лучшей точностью при многомерной оптимизации.

Также стоит заметить, что метод золотого сечения работает корректно только для унимодальных функций, а сеточный поиск практически всегда выведет приемлемый результат для любой функции. Проблема в том, что данный метод может просто пропустить искомое значение.

# **ВЫВОД:**

Мы изучили методы параметрической оптимизации, их алгоритмы, достоинства и недостатки, а также реализовали два из них.

Как мы могли заметить при сравнении методов результаты золотого сечения и сеточного поиска являются довольно-таки точными. Сложность алгоритмов заключается в том, что проводится слишком много вычислений, т.е. увеличивается время поиска значения. Также результат методов зависит от значения погрешности. Если она будет достаточно малым числом, то экстремумы, найденными нашими методами, будут практически совпадать с реальными значения. Иначе точность результатов будет отличаться от настоящих экстремумов на то количество знаков, сколько их у погрешности.

# **ПРИЛОЖЕНИЕ А:**

## **ЛИСТИНГ ПРОГРАММЫ ЗОЛОТОГО СЕЧЕНИЯ:**

#include <iostream>

#include <math.h>

using namespace std;

double f1(double x);

double f2(double x);

int main()

{

double a, eps, b, x1, x2,x3,x4;

cout << "Введите eps=";

cin >> eps;

cout << "Введите левую границу a=";

cin >> a;

cout << "Введите правую границу b=";

cin >> b;

if(a >= b)

{

a=-0.5;

b=0.5;

}

double a1,a2,b1,b2,d;

a1=a;

a2=a;

b1=b;

b2=b;

d=(sqrt(5)-1)/2;

while (b1-a1>eps)

{

x1=b1-d\*(b1-a1);

x2=a1+d\*(b1-a1);

if (f1(x1)<f1(x2))

b1=x2;

else

a1=x1;

}

while (b2-a2>eps)

{

x3=b2-d\*(b2-a2);

x4=a2+d\*(b2-a2);

if (f2(x3)<f2(x4))

b2=x4;

else

a2=x3;

}

x1 = (a1+b1)/2;

x3 = (a2+b2)/2;

cout << "Координаты минимума функции sin(x)+18x^2 на интервале" << "[" << a << ";" << b << "]: " << "( " << x1 << " , " << f1(x1) << " )" << endl;

cout << "Координаты минимума функции x^4-6x^2+10 на интервале" << "[" << a << ";" << b << "]: " << "( " << x3 << " , " << f2(x3) << " )"<< endl;

}

double f1(double x)

{

return sin(x)+18\*x\*x;

}

double f2(double x)

{

return x\*x\*x\*x-6\*x\*x+10;

}

## **ЛИСТИНГ ПРОГРАММЫ СЕТОЧНОГО ПОИСКА:**

#include <iostream>

#include <math.h>

using namespace std;

double f1(double x);

double f2(double x);

int main()

{

double a, eps, b, x1, q, xj1, xj2, min1=1000000, min2=1000000;

int k1,k2;

cout << "Введите eps=";

cin >> eps;

cout << "Введите левую границу a=";

cin >> a;

cout << "Введите правую границу b=";

cin >> b;

if(a >= b)

{

a=-0.5;

b=0.5;

}

q=(b-a)/eps;

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

{

x1=a+j\*(b-a)/q;

if (f1(x1)<min1)

{

xj1=x1;

min1=f1(x1);

}

if(f2(x1)<min2)

{

xj2=x1;

min2=f2(x1);

}

}

cout << "Координаты минимума функции sin(x)+18x^2 на интервале" << "[" << a << ";" << b << "]: " << "( " << xj1 << " , " << min1 << " )" << endl;

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

{

x1=a+j\*(b-a)/q;

if(f2(x1) == min2)

cout << "Координаты минимума функции x^4-6x^2+10 на интервале" << "[" << a << ";" << b << "]: " << "( " << x1 << " , " << min2 << " )"<< endl;

}

}

double f1(double x)

{

return sin(x)+18\*x\*x;

}

double f2(double x)

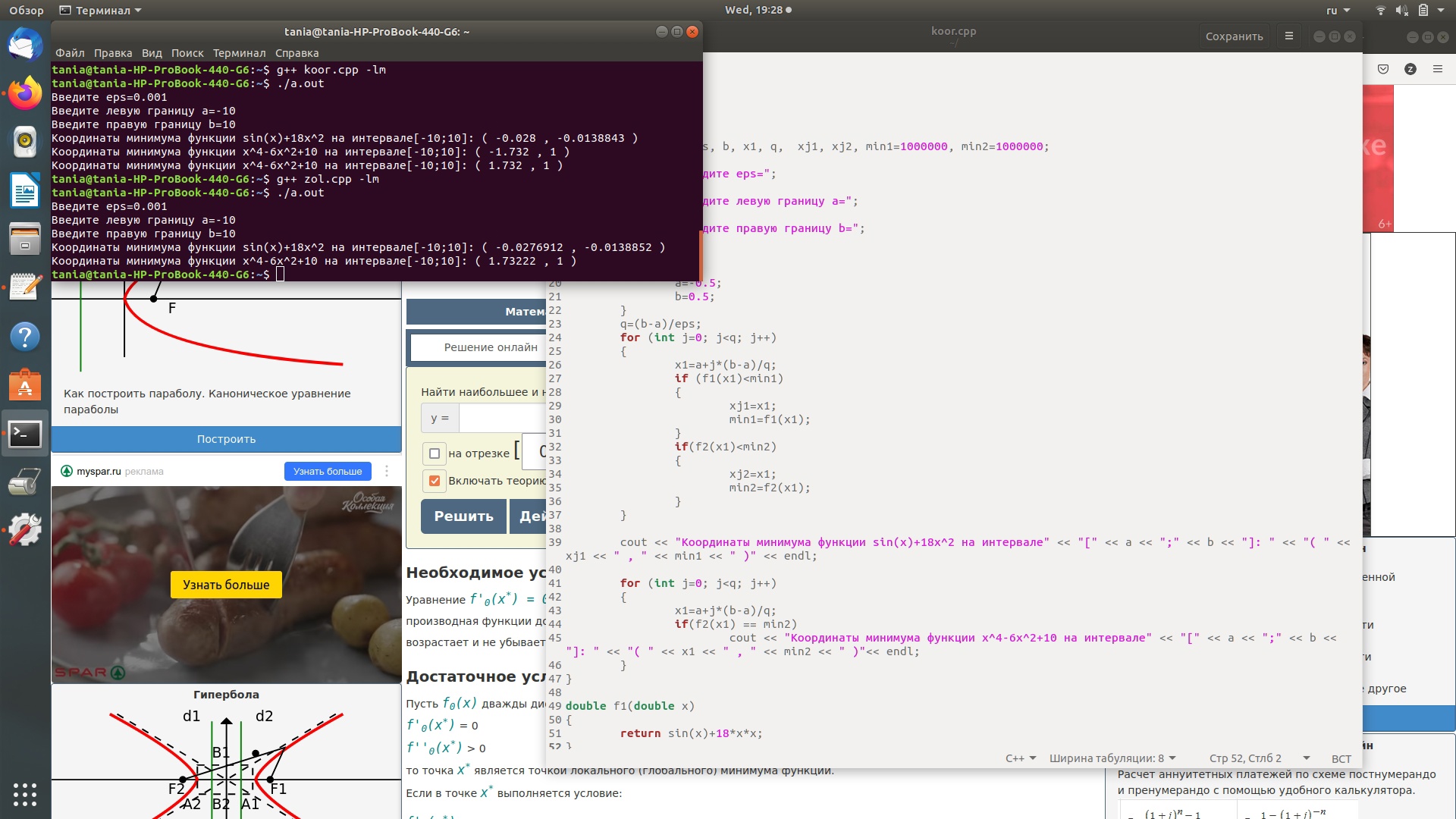
{

return x\*x\*x\*x-6\*x\*x+10;

}

# **ПРИЛОЖЕНИЕ Б:**

## **РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ПРОГРАММЫ ЗОЛОТОГО СЕЧЕНИЯ:**



## **РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ПРОГРАММЫ СЕТОЧНОГО ПОИСКА:**

