**Национальный исследовательский технологический университет «МИСиС»**

**Институт Информационных технологий и компьютерных наук (ИТКН)**

**Курс «Методы оптимизации»**

Лабораторная работа № 2

по теме

«Численные методы одномерной минимизации

с использованием производной»

Вариант №23

Выполнил:

Студент группы БИВТ-20-1

Смирнов А.А.

Проверил:

к.ф.-м.н., доцент, Лычев А.В.

Москва, 2023

Цель: приобретение практических навыков для решения задач одномерной минимизации численными методами с использованием производной.

# Ход работы:

## Формулировка

Вариант задания – №23.

Рассматриваемая функция –

Требуется найти безусловный минимум функции одной переменной y = f(x) на отрезке [a, b], где функция является унимодальной. То есть найти такую точку .

1. Графическое представление функции

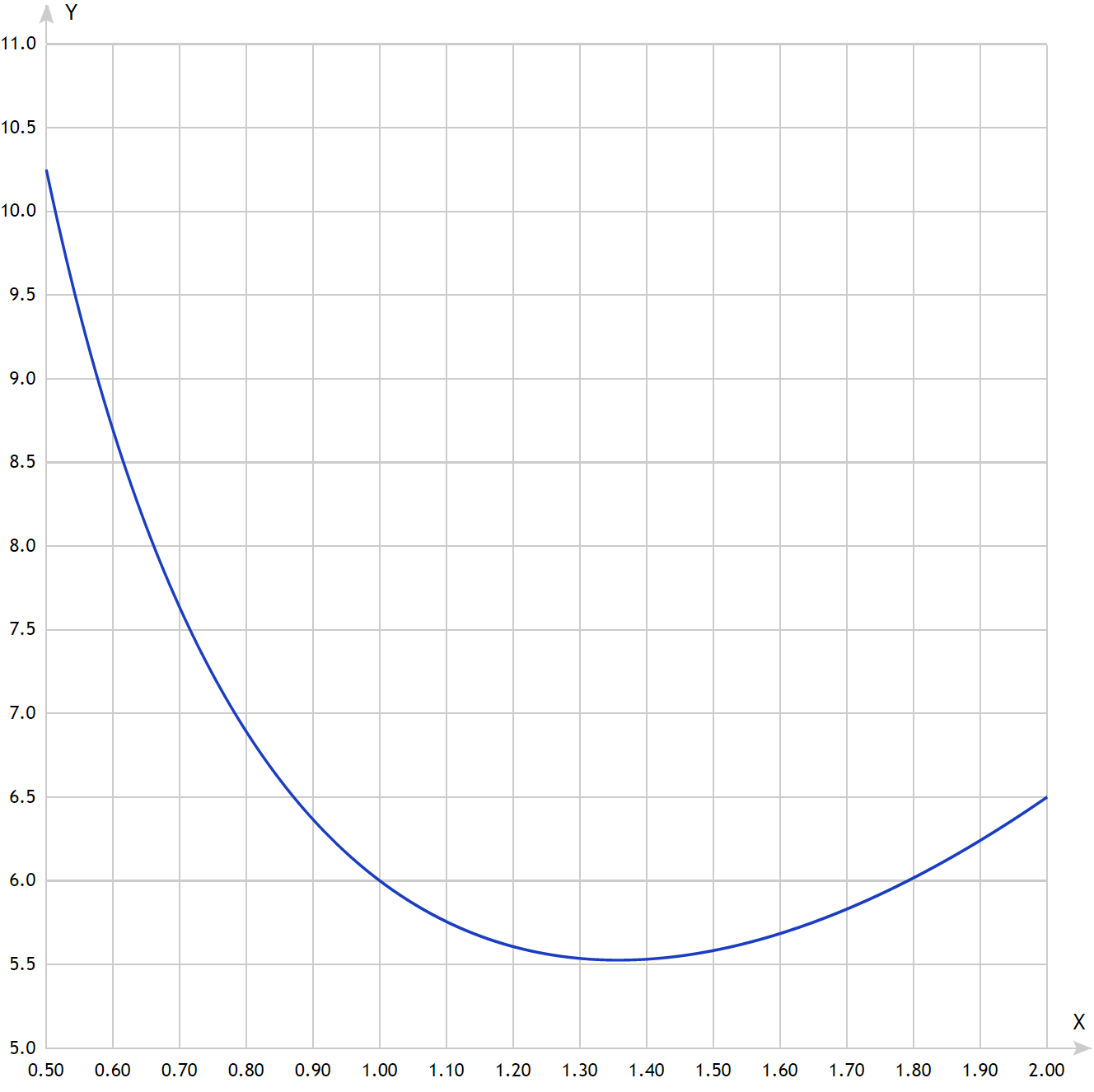


Рисунок 2.1 – График рассматриваемой функции на заданном отрезке.

## Листинг программ

Для выполнения данной лабораторной работы я реализовал несколько функций.

3.1 Функция для поиска значения исследуемой функции

def f(x):

return 5/x + x\*x

3.2 Функция для поиска значения производной исследуемой функции

def f\_1(x):

return -5/(x\*x) + 2\*x

3.3 Функция реализующая метод средней точки

def the\_midpoint\_method(a, b, e, iter = 0):

iter += 1

x = (a+b)/2

f\_1\_value = f\_1(x)

if(abs(f\_1\_value) <= e):

return [(round(x, round\_num), round(f(x), round\_num)), iter]

if(f\_1\_value > 0):

return the\_midpoint\_method(a, x, e, iter)

return the\_midpoint\_method(x, b, e, iter)

3.4 Функция реализующая метод хорд

def the\_chord\_method(a, b, e):

iter = 0

f\_1\_a\_value = f\_1(a)

f\_1\_b\_value = f\_1(b)

f\_1\_x\_value = 0

if(f\_1\_a\_value \* f\_1\_b\_value < 0):

def inner\_counting():

nonlocal f\_1\_a\_value, f\_1\_b\_value, f\_1\_x\_value, a, b, e, iter

iter += 1

x = a - f\_1\_a\_value / (f\_1\_a\_value - f\_1\_b\_value) \* (a - b)

f\_1\_x\_value = f\_1(x)

if(abs(f\_1\_x\_value) <= e):

return [(round(x, round\_num), round(f(x), round\_num)), iter]

if(f\_1\_x\_value > 0):

b = x

f\_1\_b\_value = f\_1\_x\_value

return inner\_counting()

a = x

f\_1\_a\_value = f\_1\_x\_value

return inner\_counting()

return inner\_counting()

if(f\_1\_a\_value > 0 and f\_1\_b\_value > 0):

return [(round(a, round\_num), round(f(a), round\_num)), iter]

if(f\_1\_a\_value < 0 and f\_1\_b\_value < 0):

return [(round(b, round\_num), round(f(b), round\_num)), iter]

if(f\_1\_a\_value == 0):

return [(round(a, round\_num), round(f(a), round\_num)), iter]

if(f\_1\_b\_value == 0):

return [(round(b, round\_num), round(f(b), round\_num)), iter]

return 0

## Результаты вычислений

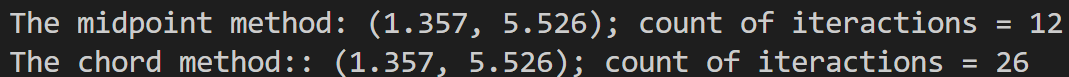


Рисунок 4.1 – Результаты вычислений минимума разными способами.

## Сравнительная характеристика методов

Таблица 6.1 – Сравнительная характеристика методов.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название метода | Число итераций | Количество вычислений функций | Найденное решение | Значение функции |
| Метода средней точки | 12 | 12 | 1.357 | 5.526 |
| Метод хорд | 26 | 28 | 1.357 | 0. 0000000 |

По данной таблице видно, что при вычислении минимума методом средней точки число итераций и вычислений функции или производной меньше, чем у метода хорд.

Вывод: ывф