Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 «Программная инженерия» –

Системное и прикладное программное обеспечение

**Отчёт**

**По лабораторной работе №2**

**По методам оптимизации**

**Вариант: 4**

Выполнил:

студент 2 курса

Батманов Даниил Евгеньевич

Группа: Р3207

Приняла:

Селина Елена Георгиевна

Отчёт принят «\_\_»\_\_\_\_\_2024 г.

Оценка: \_\_\_\_\_\_\_\_\_\_\_

г. Санкт-Петербург, 2024

**Оглавление**

[Задание 3](#_Toc160626351)

[Ручное решение 3](#_Toc160626353)

[Исходный код программы 4](#_Toc160626354)

[Результат работы программы 6](#_Toc160626355)

[Вывод 6](#_Toc160626356)

# 

# 

# 

# Задание

# Найти минимум функции на отрезке [a, b] = [0, 1] с заданной точностью 0,03 следующими методами:

* Половинное деление;
* Золотое сечение;
* Метод хорд;
* Метод Ньютона.

Написать программный продукт на одном из языков программирования для вычисления минимума и отразить первые 5 шагов каждого метода руками.

# Ручное решение

*Метод половинного деления:*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№ шага** | **a** | **b** | **x1** | **x2** | **y1** | **y2** | **b - a** |
| **1** | 0 | 1 | 0,485 | 0,515 | -0,3486 | -0,35599 | 1 |
| **2** | 0,485 | 1 | 0,7275 | 0,7575 | -0,4004 | -0,4002 | 0,515 |
| **3** | 0,485 | 0,7575 | 0,60625 | 0,63625 | -0,386 | -0,3917 | 0,2725 |
| **4** | 0,60625 | 0,7575 | 0,6669 | 0,6969 | -0,3962 | -0,399 | 0,151125 |
| **5** | 0,6669 | 0,7575 | 0,6972 | 0,7272 | -0,399 | -0,4004 | 0,0906 |

*Метод золотого сечения:*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№ шага** | **a** | **b** | **x1** | **x2** | **y1** | **y2** | **b - a** |
| **1** | 0 | 1 | 0,382 | 0,618 | -0,2998 | -0,3884 | 1 |
| **2** | 0,382 | 1 | 0,618 | 0,763924 | -0,3884 | -0,3999 | 0,618 |
| **3** | 0,618 | 1 | 0,763924 | 0,854076 | -0,3999 | -0,3892 | 0,382 |
| **4** | 0,618 | 0,854 | 0,0557 | 0,763924 | -0,054 | -0,399 | 0,236076 |
| **5** | 0,0557 | 0,854 | 0,7639 | 0,54910 | -0,399 | -0,371 | 0,798344 |

*Метод хорд:*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№ шага** | **a** | **b** | **f'(a)** | **f'(b)** | **x** | **f'(x)** | **| f'(x)|** |
| **1** | 0 | 1 | -1 | 0,45969 | 0,6850734 | -0,0892 | 0,08929928 |
| **2** | 0,6850 | 1 | -0,0892 | 0,45969 | 0,736299 | -0,0046 | 0,00466004 |
| **3** | 0,736299 | 1 | -0,004 | 0,459697 | 0,7389454 | -0,000233 | 0,00023393 |

*Метод Ньютона:*

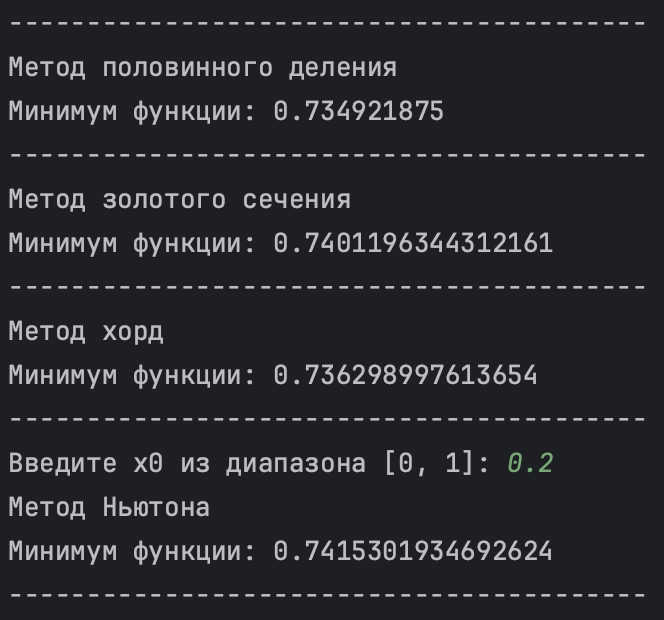
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ шага** | **x** | **f'(a)** | **f'(b)** | **| f'(x)|** |
| **1** | 0,2 | -0,7800666 | 1,19866933 | 0,78006658 |
| **2** | 0,85077712 | 0,19137801 | 1,75179307 | 0,19137801 |
| **3** | 0,74153019 | 0,00409429 | 1,67541712 | 0,00409429 |
| **4** | 0,73908645 | 2,2036E-06 | 1,673613 | 2,2036E-06 |

# Исходный код программы

import math  
  
  
a = 0  
b = 1  
epsilon = 0.03  
  
  
# исходная функция  
def f(x):  
 return 1 / 2 \* x \*\* 2 - math.sin(x)  
  
  
# половинное деление  
def halving(a\_1m, b\_1m, epsilon\_1m):  
 a\_copy = a\_1m  
 b\_copy = b\_1m  
  
 while True:  
 x1 = (a\_copy + b\_copy - epsilon\_1m) / 2  
 x2 = (a\_copy + b\_copy + epsilon\_1m) / 2  
 y1 = f(x1)  
 y2 = f(x2)  
  
 if y1 > y2:  
 a\_copy = x1  
 else:  
 b\_copy = x2  
  
 if b\_copy - a\_copy <= 2 \* epsilon\_1m:  
 return (a\_copy + b\_copy) / 2  
  
  
# золотое сечение  
def golden\_ratio(a\_2m, b\_2m, epsilon\_2m):  
 a\_copy = a\_2m  
 b\_copy = b\_2m  
  
 x1 = a\_copy + 0.382 \* (b\_copy - a\_copy)  
 x2 = a\_copy + 0.618 \* (b\_copy - a\_copy)  
 y1 = f(x1)  
 y2 = f(x2)  
  
 while True:  
 if y1 < y2:  
 b\_copy = x2  
 x2 = x1  
 y2 = y1  
 x1 = a\_copy + 0.382 \* (b\_copy - a\_copy)  
 y1 = f(x1)  
 else:  
 a\_copy = x1  
 x1 = x2  
 y1 = y2  
 x2 = a\_copy + 0.618 \* (b\_copy - a\_copy)  
 y2 = f(x2)  
  
 if abs(b\_copy - a\_copy) < epsilon\_2m:  
 return (a\_copy + b\_copy) / 2  
  
  
def d1(x):  
 return x - math.cos(x)  
  
  
def chord\_method(a\_3m, b\_3m, epsilon\_3m):  
 a\_copy = a\_3m  
 b\_copy = b\_3m  
  
 while True:  
 x\_tilda = a\_copy - (d1(a\_copy) / (d1(a\_copy) - d1(b\_copy))) \* (a\_copy - b\_copy)  
 d1\_x\_tilda = d1(x\_tilda)  
  
 if abs(d1\_x\_tilda) <= epsilon\_3m:  
 return x\_tilda  
  
 if d1\_x\_tilda > 0:  
 b\_copy = x\_tilda  
 else:  
 a\_copy = x\_tilda  
  
  
def d2(x):  
 return math.sin(x) + 1  
  
  
def newton\_method(a\_4m, b\_4m, epsilon\_4m, x0\_4m):  
 if x0\_4m >= a\_4m and x0\_4m <= b\_4m:  
 x = x0\_4m  
 else:  
 return "Некорректный x0"  
  
 while True:  
 if abs(d1(x)) <= epsilon\_4m:  
 return x  
 x -= d1(x) / d2(x)  
  
  
def separate():  
 print("-----------------------------------------")  
  
  
separate()  
print("Метод половинного деления\nМинимум функции:", halving(a, b, epsilon))  
separate()  
print("Метод золотого сечения\nМинимум функции:", golden\_ratio(a, b, epsilon))  
separate()  
print("Метод хорд\nМинимум функции:", chord\_method(a, b, epsilon))  
separate()  
print("Метод Ньютона\nМинимум функции:", newton\_method(a, b, epsilon, float(input("Введите x0 из диапазона [0, 1]: "))))  
separate()

# Результат работы программы

*Вывод:*



# Вывод

В ходе выполнения данной лабораторной работы я научился искать минимум функции методами половинного деления, золотого сечения, хорд и Ньютона; написал программные продукты для уравнения решения каждым методом.