Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет Программной инженерии и компьютерной техники

Отчёт по лабораторной работе №2

Методы оптимизации

Вариант № 9

Выполнил: студент группы P3214

Силинцев В.В.

Преподаватель: Селина Е.Г.

Содержание

[Цель работы 3](#__RefHeading___Toc520_1614521587)

[Задание 4](#__RefHeading___Toc522_1614521587)

[Ручные расчеты 5](#__RefHeading___Toc522_1614521587_%2525D0)

[Метод половинного деления 5](#__RefHeading___Toc1093_2157254880)

[Метод золотого сечения 8](#__RefHeading___Toc1093_2157254880_%2525D)

[Метод хорд 11](#__RefHeading___Toc1093_2157254880_%25251)

[Метод Ньютона 13](#__RefHeading___Toc1093_2157254880_%25252)

[Код программы 15](#__RefHeading___Toc524_1614521587_%2525D0)

[Результат работы программы 16](#__RefHeading___Toc524_1614521587_%2525D1)

[Заключение 17](#__RefHeading___Toc524_1614521587)

# **Цель работы**

Изучить четыре метода нахождения экстремума: метод половинного деления, метод золотого сечения, метод хорд, метод Ньютона и реализовать их программно.

# **Задание**

Решить задачу четырьмя методами: методом половинного деления, методом золотого сечения, методом хорд и методом Ньютона. По 5 шагов каждого метода выполнить вручную + написать программу по каждому методу на одном из языков программирования.

* Функция для нахождения экстремума: .
* Интервал для поиска: .
* Необходимая точность вычислений: для ручных расчетов и для программных расчетов.

# **Ручные расчеты**

## Метод половинного деления

Алгоритм поиска экстремума:

* Берем две точки вблизи середины интервала :   
  , .
* Вычисляем , .
* Если , тогда присваивается , иначе присваивается .
* Если , тогда повторяем с п.1, иначе переходим к пункту 5.
* Вычисляем , .

Поиск методом половинного деления (5 итераций):

Первая итерация:

1. Находим и .
2. Вычисляем и .
3. , тогда .
4. , тогда повторяем пункт 1.

Вторая итерация:

1. Находим и .
2. Вычисляем и .
3. , тогда .
4. , тогда повторяем пункт 1.

Третья итерация:

1. Находим и .
2. Вычисляем и .
3. , тогда .
4. , тогда повторяем пункт 1.

Четвертая итерация:

1. Находим и .
2. Вычисляем и .
3. , тогда .
4. , тогда повторяем пункт 1.

Пятая итерация:

1. Находим и .
2. Вычисляем и .
3. , тогда .
4. .
5. Находим , . Заканчиваем поиск.

Таблица 1. Поиск методом половинного деления (5 итераций).

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № итерации |  |  |  |  |  |  |  |
| 1 | 1.5 | 2 | 1.74 | 1.76 | -5.98023 | -5.98779 | 0.5 |
| 2 | 1.74 | 2 | 1.86 | 1.88 | -6.00078 | -5.99832 | 0.26 |
| 3 | 1.74 | 1.88 | 1.8 | 1.82 | -5.99798 | -6.00059 | 0.14 |
| 4 | 1.8 | 1.88 | 1.83 | 1.85 | -6.00127 | -6.00136 | 0.08 |
| 5 | 1.83 | 1.88 | 1.845 | 1.865 | -6.0015 | -6.00032 | 0.05 |

## Метод золотого сечения

Алгоритм поиска экстремума:

* На первом шаге (итерации) точки вычисляются по формулам: , .
* Затем вычисляются значение функции в этих точках.
* Если , то оставляем отрезок . На второй итерации полагаем равным , а вычисляем по формуле . Значение функции вычисляется только в точке , так как значение функции в уже было вычислено на предыдущем шаге.
* Если , то оставляем отрезок . На второй итерации полагаем равным , а вычисляем по формуле . Значение функции вычисляется только в точке , так как значение функции в уже было вычислено на предыдущем шаге.
* Вычисления продолжают до тех пор, пока длина интервала не станет меньше требуемой точности.

Поиск методом половинного деления (5 итераций):

Первая итерация:

1. Находим и .
2. Вычисляем и .
3. , тогда , , .
4. , тогда повторяем пункт 2.

Вторая итерация:

1. Вычисляем и .
2. , тогда , , .
3. , тогда повторяем пункт 1.

Третья итерация:

1. Вычисляем и .
2. , тогда , , .
3. , тогда повторяем пункт 1.

Четвертая итерация:

1. Вычисляем и .
2. , тогда , , .
3. , тогда повторяем пункт 1.

Пятая итерация:

1. Вычисляем и .
2. , тогда , , .
3. , заканчиваем поиск.

Таблица 2. Поиск методом золотого сечения (5 итераций).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № итерации |  |  |  |  |  |  |
| 1 | 1.5 | 2 | 1.691 | 1.809 | -5.95489 | -5.999365 |
| 2 | 1.691 | 2 | 1.809 | 1.809038 | -5.999365 | -5.999371 |
| 3 | 1.809 | 2 | 1.809038 | 1.927015 | -5.999371 | -5.985756 |
| 4 | 1.809 | 1.927015 | 1.809015 | 1.809038 | -5.999367 | -5.999371 |
| 5 | 1.809015 | 1.927015 | 1.809038 | 1.881925 | -5.999371 | -5.997992 |

## Метод хорд

Алгоритм поиска экстремума:

* Шаг 1. Находим по формуле . Вычисляем и переходим к шагу 2.
* Шаг 2. Проверка на окончание поиска: если , то положить , , и завершить поиск, иначе перейти к шагу 3.
* Шаг 3. Переход к новому отрезку. Если , то положить , , иначе положить , . Перейти к шагу 1.

Поиск методом хорд (3 итерации):

Первая итерация:

1. Найдем производную .
2. Находим . Вычисляем .
3. , тогда продолжаем поиск.
4. , тогда , . Повторяем пункт 2.

Вторая итерация:

1. Находим . . Вычисляем .
2. , продолжаем поиск для большей точности.
3. , тогда , . Повторяем пункт 1.

Третья итерация:

1. Находим . . Вычисляем .
2. , завершаем поиск.

Таблица 3. Поиск методом хорд (3 итерации).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № итерации |  |  |  |  |  |  |
| 1 | 1.5 | 2 | 1.829918 | -1.344535 | 0.693147 | -0.047130 |
| 2 | 1.829918 | 2 | 1.840746 | -0.047130 | 0.693147 | -0.001483 |
| 3 | 1.840746 | 2 | 1.841086 | -0.001483 | 0.693147 | -0.000046 |

## Метод Ньютона

Алгоритм поиска экстремума:

* Выбираем начальное приближение к искомой точке .
* Выберем в качестве следующего приближения к точку .
* Далее продолжаем находить приближения по формуле .
* Продолжаем вычисления пока не будет выполнено .

Поиск методом Ньютона (3 итерации):

Первая итерация:

1. Найдем производную .
2. Найдем вторую производную .
3. Выберем начальное приближение .
4. В качестве следуещего приближения возьмем . .
5. , тогда продолжим поиск. Повторяем пункт 4.

Вторая итерация:

1. В качестве следуещего приближения возьмем . .
2. , тогда продолжим поиск. Повторяем пункт 1.

Третья итерация:

1. В качестве следуещего приближения возьмем . .
2. , тогда завершаем поиск.

Таблица 4. Поиск методом Ньютона (3 итерации).

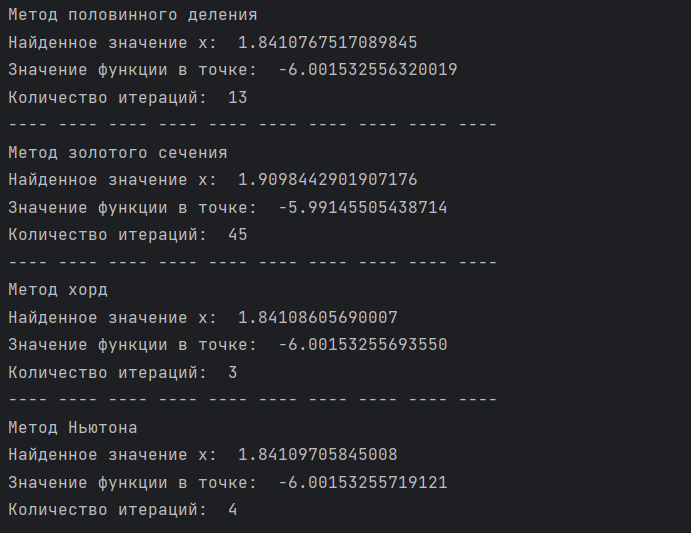
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № итерации, |  |  |  |  |
| 1 | 1.5 | 1.867 | -1.34 | 3.667 |
| 2 | 1.867 | 1.841 | 0.11 | 4.269 |
| 3 | 1.841 | 1.841 | 0 | 4.226 |

# **Код программы**

Полный исходный код приложения:

[https://github.com/vvlaads/vvlaads/tree/master/Optimization%20methods/Lab2](https://github.com/vvlaads/vvlaads/tree/master/Optimization methods/Lab2)

# **Результат работы программы**

Рисунок 1: Результат работы программы.

# **Заключение**

В ходе этой работы я познакомился с методами поиска экстремума: методом половинного деления, методом золотого сечения, методом хорд и методом Ньютона. Я научился применять их на практике, а также реализовал их программно.

Наиболее быстрыми методами поиска оказались метод Ньютона и метод Хорд. Самым медленным оказался метод золотого сечения. Наиболее точным из методов оказался метод Ньютона.