ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Факультет информационных технологий и программирования

Дисциплина:

«Прикладная математика»

Отчёт по лабораторной работе №2

«Методы одномерного поиска»

Группа M32001

Выполнили:

Соловьев Роман Сергеевич

Халилов Роман Эдуардович

Преподаватель:

Москаленко Мария Александровна

Санкт-Петербург

2023 г.

Цель работы:
Анализ методов одномерной минимизации унимодальной на интервале функции без производной.

Постановка задачи:
Необходимо реализовать методы нахождения минимума унимодальной функции на интервале, не использующие производную. Сравнить методы по количеству итераций и количеству вычислений функции в зависимости от разных функций. Применить алгоритмы на различных многомодальных функциях.

Ход работы:
Весь код, использованный в ходе выполнения работы: GitHub

Вариант 5.

Часть 1 – выполнение алгоритмов на заданной функции:

Для исследования методов используем функцию на отрезке [-5.1, 2.45]:

$f_1(x) = e^{\sin(x)} * x^2$,

Метод 1: метод дихотомии

На каждой итерации алгоритма выбираются две точки внутри отрезка, находящиеся на одинаковом расстоянии от его центра, при этом данное расстояние меньше, чем половина длины возвращаемого отрезка. Сравниваются значения функций в этих точках, на основании чего один из образуемых подотрезков выбрасывается и производится следующая итерация алгоритма на новом отрезке, пока его длинна не будет меньше запрашиваемой точности.
Исходя из этого на каждой итерации алгоритма интервал неопределённости уменьшается примерно в 2 раза.

За n итераций длина интервала $l \approx \frac{b_0 - a_0}{2^n}$, где $n(\varepsilon) \approx \frac{ln\left(\frac{b_0 - a_0}{\varepsilon}\right)}{ln\,2}$ – количество итераций, количество вычислений функций m = 2* $n(\varepsilon)$

h	l	Iter_cnt	Func_calc_cnt
0.16	0.138	7	14
0.08	0.069	8	16
0.04	0.035	9	18
0.02	0.017	10	20
0.01	0.009	11	22

Изменение отрезка для каждого исследуемого h есть в коде, в отчёте будут приведены только для h = 0.04

Iter_num	a_i	b_i
0	-5.1	2.45
1	-1.335	2.45
2	-1.335	0.568
3	-0.394	0.568
4	-0.394	0.097
5	-0.158	0.097
6	-0.041	0.097
7	-0.041	0.038
8	-0.011	0.038
9	-0.011	0.023

Метод 2: метод золотого сечения

На каждой итерации алгоритма выбираются две точки внутри отрезка, находящиеся на одинаковом расстоянии от его центра, при этом они делят данный отрезок в пропорции золотого сечения. Сравниваются значения функций в этих точках, на основании чего один из образуемых подотрезков выбрасывается и производится следующая итерация

алгоритма на новом отрезке, пока его длинна не будет меньше запрашиваемой точности. При этом одна из новых точек разбиения отрезка будет совпадать с предыдущей точкой, поэтому на каждой новой итерации нужно будет вычислить только одно значение функции.
Исходя из этого на каждой итерации алгоритма интервал неопределённости уменьшается примерно в 1.618 раз.

За n итераций длина интервала $l \approx \frac{b_0 - a_0}{1.618^n}$, где $n(\varepsilon) \approx \frac{ln\left(\frac{b_0 - a_0}{\varepsilon}\right)}{\ln\left(\frac{\sqrt{5}-1}{2}\right)}$ – количество итераций, количество вычислений функций m = $n(\varepsilon) + 2$

h	l	Iter_cnt	Func_calc_cnt
0.16	0.099	9	11
0.08	0.061	10	12
0.04	0.038	11	13
0.02	0.014	13	15
0.01	0.009	14	16

Изменение отрезка для h = 0.04

Iter_num	a_i	b_i
0	-5.1	2.45
1	-2.216	2.45
2	-2.216	0.668
3	-1.115	0.668
4	-0.434	0.668
5	-0.434	0.247
6	-0.174	0.247
7	-0.174	0.086
8	-0.074	0.086
9	-0.074	0.025
10	-0.037	0.025
11	-0.013	0.025

Метод 3: метод Фибоначчи

На каждой итерации алгоритма выбираются две точки внутри интервала, при этом они находятся на расстоянии $\frac{F_n}{F_{n+2}} \cdot l$ и $\frac{F_{n+1}}{F_{n+2}} \cdot l$ от его левой границы. Сравниваются значения функций в этих точках, на основании чего один из образуемых подотрезков выбрасывается и производится следующая итерация алгоритма на новом отрезке с меньшими номерами чисел Фибоначчи, пока его длинна не будет меньше запрашиваемой точности. При этом одна из новых точек разбиения отрезка будет совпадать с предыдущей точкой, поэтому на каждой новой итерации нужно будет вычислить только одно значение функции.

Исходя из этого на каждой итерации алгоритма интервал неопределённости уменьшается примерно в 1.618 раз.
За n итераций длина интервала $l \approx \frac{b_0 - a_0}{F_{n+2}}$, где $n(\varepsilon)$: $F_{n+2} > \frac{b_0 - a_0}{\varepsilon}$ – количество итераций, количество вычислений функций m = $n(\varepsilon)\ + 2$

h	l	Iter_cnt	Func_calc_cnt
0.16	0.105	9	11
0.08	0.040	11	13
0.04	0.025	12	14
0.02	0.009	14	16
0.01	0.006	15	17

Изменение отрезка для h = 0.04

Iter_num	a_i	b_i
0	-5.1	2.45
1	-2.216	2.45
2	-2.216	0.668
3	-1.115	0.668
4	-0.434	0.668
5	-0.434	0.247
6	-0.174	0.247
7	-0.174	0.086
8	-0.075	0.086
9	-0.075	0.024
10	-0.038	0.024
11	-0.013	0.024
12	-0.013	0.012

Метод 4: метод парабол

На каждой итерации алгоритма выбираются две точки внутри отрезка, при этом одна из них выбирается как точка минимума параболы, проходящей через другую выбранную точку и края отрезка. Сравниваются значения функций в этих точках, на основании чего один из образуемых подотрезков выбрасывается и производится следующая итерация алгоритма на новом отрезке, пока его длинна не будет меньше запрашиваемой точности. При этом одна из новых точек разбиения отрезка будет, которая не является минимумом новой параболы, будет одной из двух точек предыдущего шага. Тогда вычисление значения функции на каждой итерации будет производиться только один раз.
Количество вычислений функций m = $n(\varepsilon)\ +\ 3$

h	l	Iter_cnt	Func_calc_cnt
0.16	0.058	5	8
0.08	0.058	5	8
0.04	0.030	6	9
0.02	0.015	183	186
0.01	0.007	184	187

Изменение отрезка для h = 0.04

Iter_num	a_i	b_i
0	-5.1	2.45
1	-1.325	2.45
2	-0.469	2.45
3	-0.138	2.45
4	-0.138	0.029
5	-0.029	0.029
6	-0.0005	0.029

Метод 5: комбинированный метод Брента

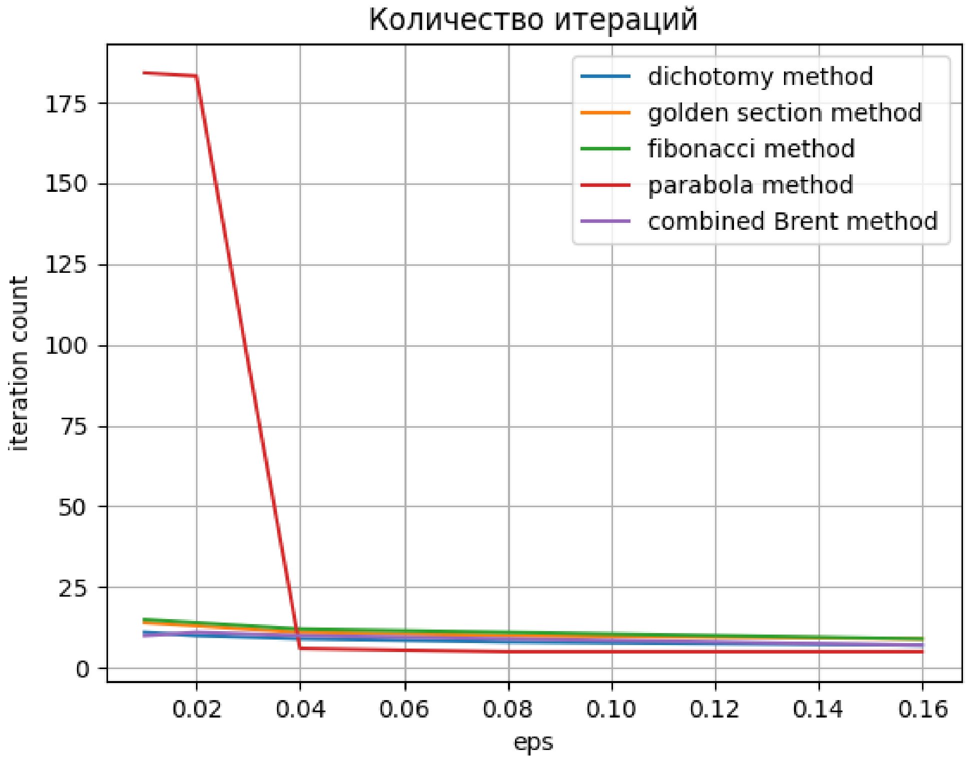
На каждой итерации алгоритма выбираются две точки внутри отрезка по принципу метода парабол. При этом если расстояние минимума параболы и любого из краёв текущего интервала неопределённости меньше необходимой точности, или если минимум параболы находится вне текущего отрезка, то используется метод золотого сечения. Сравниваются значения функций в этих точках, на основании чего один из образуемых подотрезков выбрасывается и производится следующая итерация алгоритма на новом отрезке, пока его длинна не будет меньше запрашиваемой точности.
Количество вычислений функций m = $n(\varepsilon) + 1$, так как оба используемых метода пересчитывают функцию только для одной точки

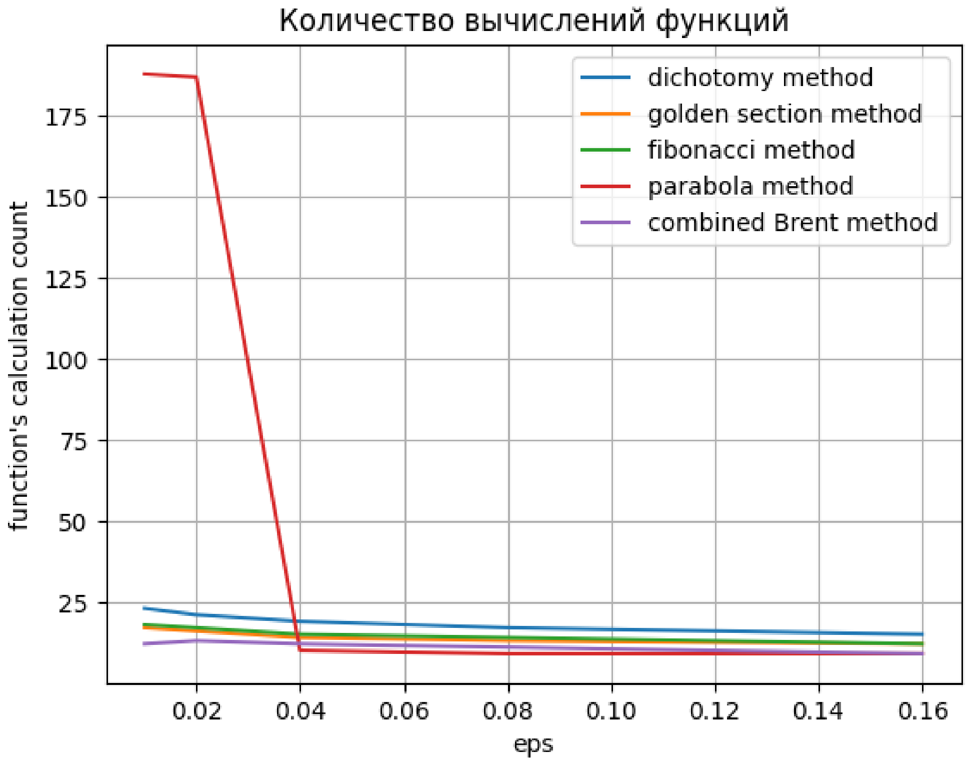
h	l	Iter_cnt	Func_calc_cnt
0.16	0.105	7	8
0.08	0.074	9	10
0.04	0.006	10	11
0.02	0.006	11	12
0.01	0.007	10	11

Изменение отрезка для h = 0.04

Iter_num	a_i	b_i
0	-5.1	2.45
1	-2.767	2.45
2	-1.325	2.45
3	-0.369	2.45
4	-0.369	0.147
5	-0.369	0.117
6	-0.165	0.117
7	-0.039	0.117
8	-0.009	0.117
9	-0.009	0.040
10	-0.009	0.010

Сравнение количества итераций методов и количества вычислений значений функции методов от точности вычисления:



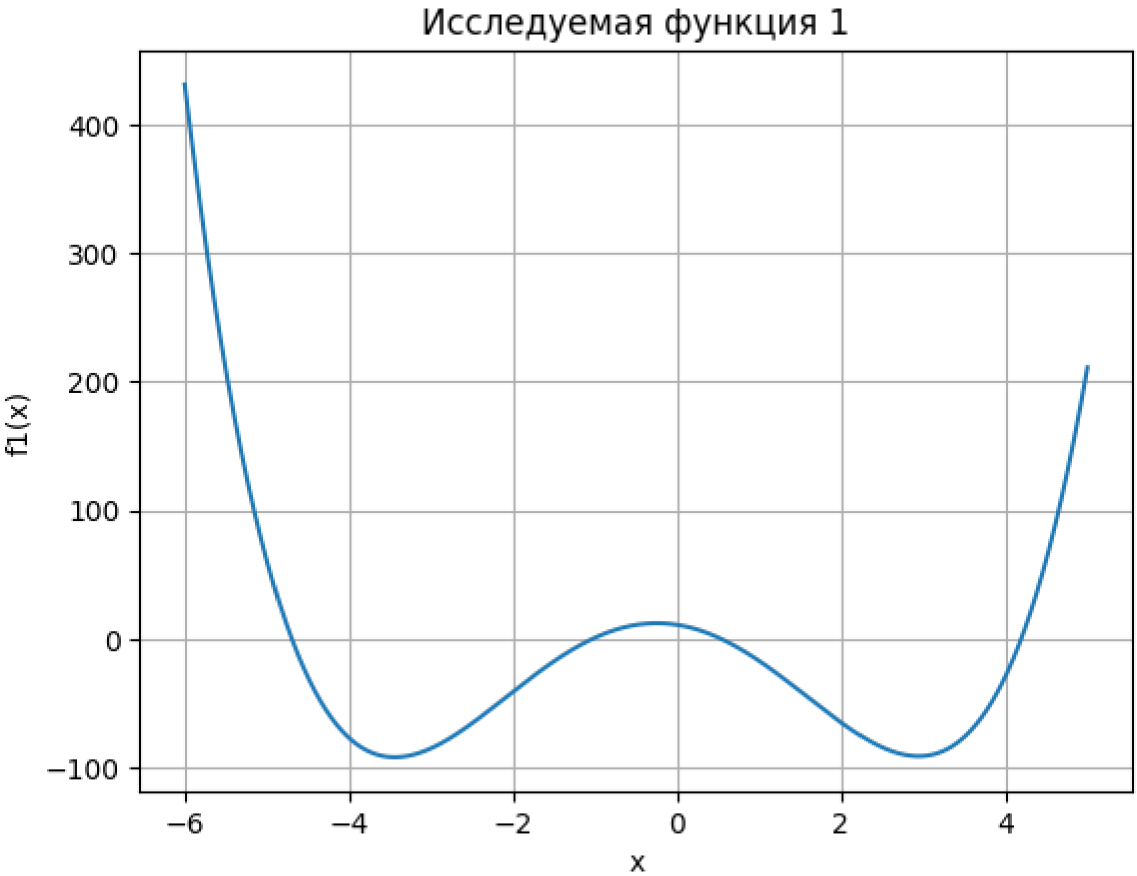


Вывод:

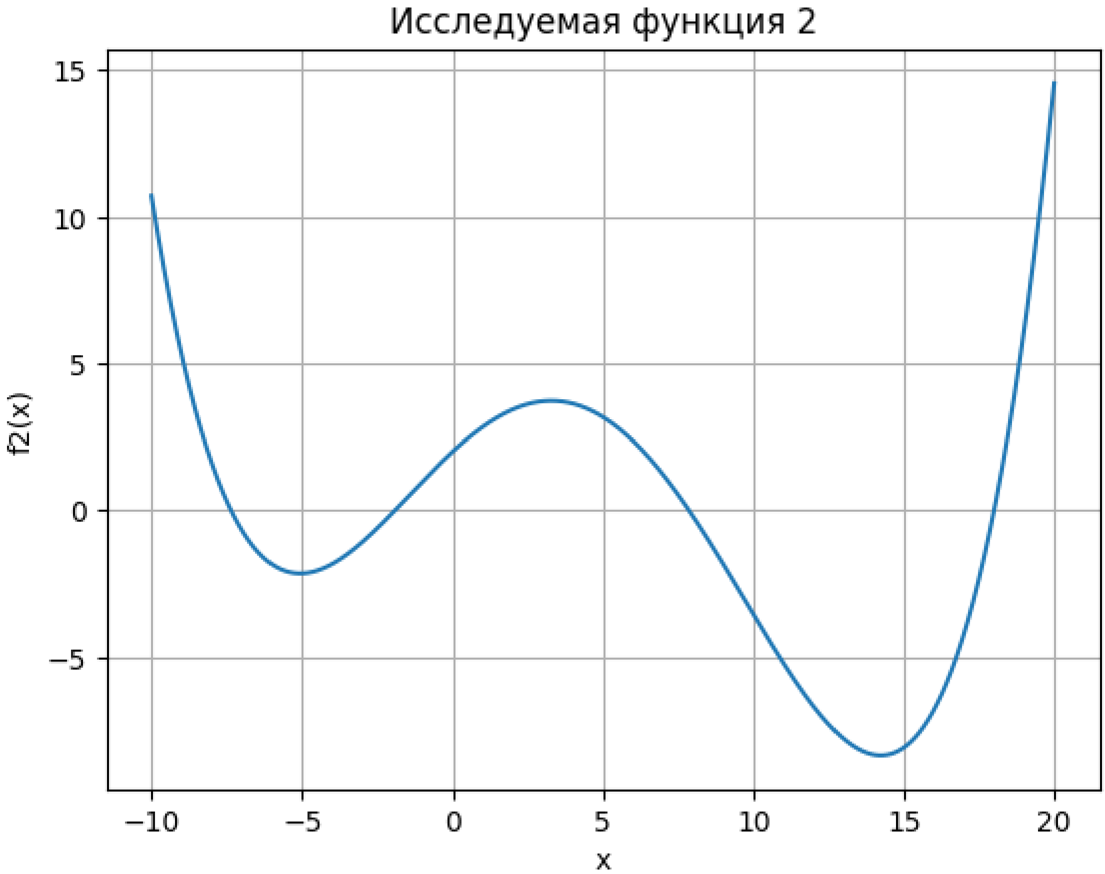
Вычислив минимум унимодальной функции с помощью разных методов одномерного поиска, сравнили их между собой по метрике количества итераций и метрике количества вычислений значения функции, наиболее эффективным в обоих случаях оказался комбинированный метод Брента.

Часть 2 – минимизация многомодальных функций:
Для исследования используем следующие 3 функции:

- $f1(x) = x^4 + x^3 - 20x^2 - 10x + 11$
  на отрезке [-6, 5]



- $f2(x) = 0.001x^4 - 0.0166x^3 - 0.084x^2 + 0.948x + 2$
  на отрезке [-10, 20]



- $f3(x) = 0.00001x^6 - 0.0042x^4 + 0.003x^3 + 0.374x^2 - 0.452x - 0.329$ на отрезке [-20, 19]

Тогда отрезки для точности вычисления eps = 0.04 (для остальных eps таблицы есть в коде)

- Для f_1: x_{min} = -3.4433

	a	b
dichotomy method	-3.4587	-3.4280
golden section method	-3.4585	-3.4244
fibonacci method	-3.4590	-3.4368
parabola method	2.9403	2.9403
combined Brent method	-3.4501	-3.4146

- Для f_2: x_{min} = 14.2311

	a	b
dichotomy method	14.2124	14.2470
golden section method	14.2171	14.2527
fibonacci method	14.2183	14.2415
parabola method	14.2311	14.2311
combined Brent method	-5.0881	-5.0486

- Для f_3: x_{min} = -15.3148

	a	b
dichotomy method	14.6024	14.6414
golden section method	0.5867	0.6153
fibonacci method	0.5867	0.6169
parabola method	0.6048	0.6048
combined Brent method	0.6003	0.6339

Вывод:

Воспользовавшись методами на многомодальных функциях, получаем локальные минимумы, не обязательно совпадающие с фактическими. Из этого следует, что условие на унимодальность функции – необходимое условие для каждого из методов.

Общий вывод по проделанной работе:

В ходе работы были реализованы методы одномерного поиска нулевого порядка, исправно работающие на унимодальных функциях, и произведено их сравнение по двум критериям – количество итераций и количество вычислений значения функции от требуемой точности. Было подтверждено, использованием методов на многомодальных функциях, что для них методы возвращают один локальный минимум, которому не гарантируется минимальное значение на рассматриваемом интервале.