

Отчёт по лабораторной работе №1
Вариант №6

Ермишина Мелисса, Тукалло Мария, Скрипченко Мария

Февраль 2021

Содержание

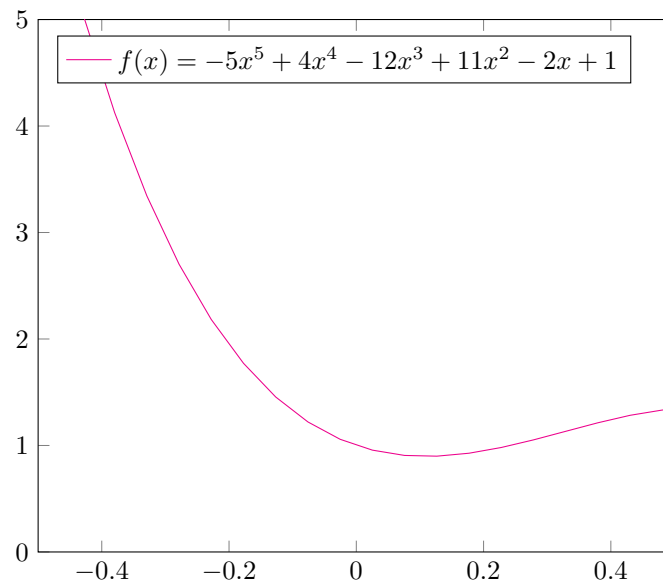
1	Цель работы	3
2	Ход работы	3
2.1	Аналитическое решение	3
2.2	Дихотомия	4
2.2.1	Вывод	4
2.3	Метод золотого сечения	5
2.3.1	Вывод	6
2.4	Метод Фибоначчи	7
2.4.1	Вывод	7
2.5	Метод парабол	9
2.5.1	Вывод	9
2.6	Метод Брента	10
2.6.1	Вывод	10
3	Тестирование алгоритмов на многомодальной функции	11
4	Результаты исследования	11

1 Цель работы

- Реализация и последующее тестирование алгоритмов одномерной минимизации функций. Таких как:
 1. Метод дихотомии
 2. Метод золотого сечения
 3. Метод Фибоначи
 4. Метод парабол
 5. Комбинированный метод Брента
- Сравнение результатов и погрешностей, предоставляемых каждым алгоритмом на конкретной функции. Сравнение их с результатами аналитического решения.

2 Ход работы

2.1 Аналитическое решение



Дана функция $f(x) = -5x^5 + 4x^4 - 12x^3 + 11x^2 - 2x + 1$.

Задача: найти $\min_{[-0.5; 0.5]} f(x)$ аналитическим методом.

Найдём производную: $f'(x) = -25x^4 + 16x^3 - 36x^2 + 22x - 2$

Найдём критические точки, приравняв производную к нулю:

$$\begin{aligned} -25x^4 + 16x^3 - 36x^2 + 22x - 2 &= 0 \\ x &\approx 0.10986 \end{aligned}$$

Итого, одна критическая точка.

Проверим знак производной на отрезках $[-0.5; 0.10986]$ и $(0.10986, 0.5]$:

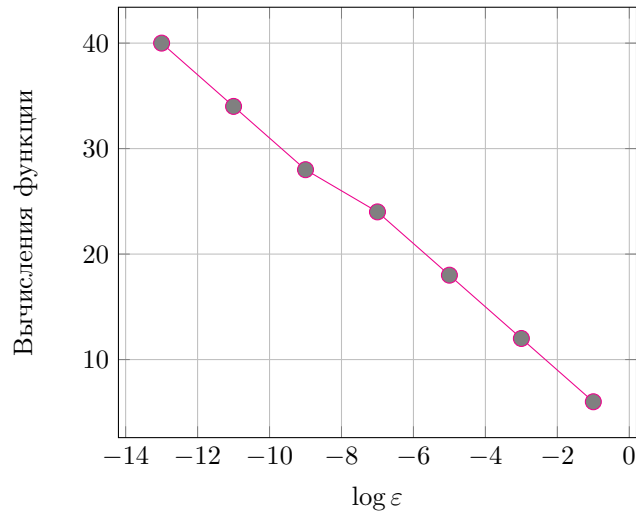
1. $f'(x) < 0$, при $x \in [-0.5; 0.10986]$
2. $f'(x) > 0$, при $x \in (0.10986, 0.5]$

Следовательно точка $x^* \approx 0.10986$ - точка минимума.

2.2 Дихотомия

№	left	right	length	x	$f(x)$
0	-0.5000000000	0.5000000000	1.0000000000	0.0000000000	1.0000000000
1	-0.0000000005	0.5000000000	0.5000000005	0.2499999997	1.0107421871
2	-0.0000000005	0.2500000003	0.2500000008	0.1249999999	0.8992614746
3	-0.0000000005	0.1250000004	0.1250000009	0.0624999999	0.9150953293
4	0.0624999994	0.1250000004	0.0625000010	0.0937499999	0.8995647729
5	0.0937499994	0.1250000004	0.0312500010	0.1093749999	0.8976346822
6	0.1093749994	0.1250000004	0.0156250010	0.1171874999	0.8980189745
7	0.1093749994	0.1171875004	0.0078125010	0.1132812499	0.8977176183
8	0.1093749994	0.1132812504	0.0039062510	0.1113281249	0.8976486056
9	0.1093749994	0.1113281254	0.0019531260	0.1103515624	0.8976347275
10	0.1093749994	0.1103515629	0.0009765635	0.1098632811	0.8976329720
11	0.1093749994	0.1098632817	0.0004882823	0.1096191405	0.8976333934
12	0.1096191400	0.1098632817	0.0002441417	0.1097412108	0.8976330743
13	0.1097412103	0.1098632817	0.0001220713	0.1098022460	0.8976329961
14	0.1098022455	0.1098632817	0.0000610362	0.1098327636	0.8976329773
15	0.1098327630	0.1098632817	0.0000305186	0.1098480223	0.8976329729
16	0.1098480218	0.1098632817	0.0000152598	0.1098556517	0.8976329720
17	0.1098556512	0.1098632817	0.0000076304	0.1098594664	0.8976329719
18	0.1098594659	0.1098632817	0.0000038157	0.1098613738	0.8976329719
19	0.1098594659	0.1098613743	0.0000019084	0.1098604201	0.8976329719
20	0.1098594659	0.1098604206	0.0000009547	0.1098599433	0.8976329719
21	0.1098594659	0.1098599438	0.0000004779	0.1098597049	0.8976329719
22	0.1098597043	0.1098599438	0.0000002394	0.1098598241	0.8976329719
23	0.1098598236	0.1098599438	0.0000001202	0.1098598837	0.8976329719
24	0.1098598832	0.1098599438	0.0000000606	0.1098599135	0.8976329719
25	0.1098598832	0.1098599140	0.0000000308	0.1098598986	0.8976329719
26	0.1098598981	0.1098599140	0.0000000159	0.1098599060	0.8976329719
27	0.1098599055	0.1098599140	0.0000000085	0.1098599097	0.8976329719
28	0.1098599092	0.1098599140	0.0000000048	0.1098599116	0.8976329719
29	0.1098599092	0.1098599121	0.0000000029	0.1098599107	0.8976329719

Зависимость числа шагов от $\log \varepsilon$



2.2.1 Вывод

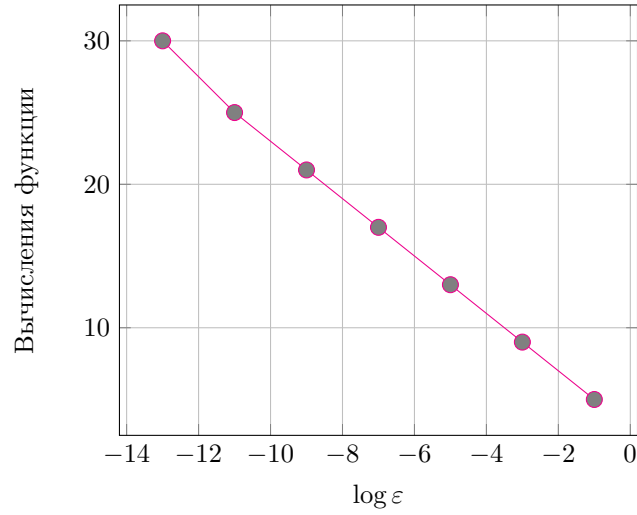
Из полученного графика видно, что количество вычислений функции для метода дихотомии линейно зависит от логарифма ε . Данный метод на каждой итерации вычисляет значение функции в двух

точках, что негативно отражается на скорости вычислений. Также нужно выбирать Δ с учетом некоторых соображений: при уменьшении Δ повышается скорость сходимости, но при чрезмерно малом Δ сравнение точек становится затруднительным. К преимуществам метода дихотомии можно отнести относительную простоту в написании.

2.3 Метод золотого сечения

№	left	right	length	x	$f(x)$
0	-0.5000000000	0.5000000000	1.0000000000	0.0000000000	1.0000000000
1	-0.1180339887	0.5000000000	0.6180339887	0.1909830056	0.9397125964
2	-0.1180339887	0.2639320225	0.3819660113	0.0729490169	0.9080836327
3	0.0278640450	0.2639320225	0.2360679775	0.1458980338	0.9065669717
4	0.0278640450	0.1737620788	0.1458980338	0.1008130619	0.8982358981
5	0.0835921350	0.1737620788	0.0901699437	0.1286771069	0.9001344998
6	0.0835921350	0.1393202250	0.0557280900	0.1114561800	0.8976514481
7	0.0835921350	0.1180339887	0.0344418537	0.1008130619	0.8982358981
8	0.0967477525	0.1180339887	0.0212862363	0.1073908706	0.8976774447
9	0.1048783713	0.1180339887	0.0131556175	0.1114561800	0.8976514481
10	0.1048783713	0.1130089900	0.0081306188	0.1089436806	0.8976390820
11	0.1079839913	0.1130089900	0.0050249987	0.1104964906	0.8976359145
12	0.1079839913	0.1110896113	0.0031056200	0.1095368013	0.8976337311
13	0.1091702326	0.1110896113	0.0019193787	0.1101299219	0.8976335016
14	0.1091702326	0.1103564738	0.0011862413	0.1097633532	0.8976330397
15	0.1096233364	0.1103564738	0.0007331374	0.1099899051	0.8976330947
16	0.1096233364	0.1100764403	0.0004531039	0.1098498883	0.8976329726
17	0.1097964067	0.1100764403	0.0002800336	0.1099364235	0.8976330144
18	0.1097964067	0.1099694770	0.0001730703	0.1098829418	0.8976329757
19	0.1097964067	0.1099033700	0.0001069633	0.1098498883	0.8976329726
20	0.1098372630	0.1099033700	0.0000661070	0.1098703165	0.8976329727
21	0.1098372630	0.1098781194	0.0000408563	0.1098576912	0.8976329719
22	0.1098528688	0.1098781194	0.0000252506	0.1098654941	0.8976329721
23	0.1098528688	0.1098684745	0.0000156057	0.1098606716	0.8976329719
24	0.1098528688	0.1098625136	0.0000096449	0.1098576912	0.8976329719
25	0.1098565528	0.1098625136	0.0000059609	0.1098595332	0.8976329719
26	0.1098588296	0.1098625136	0.0000036840	0.1098606716	0.8976329719
27	0.1098588296	0.1098611065	0.0000022768	0.1098599681	0.8976329719
28	0.1098588296	0.1098602368	0.0000014072	0.1098595332	0.8976329719
29	0.1098593671	0.1098602368	0.0000008697	0.1098598020	0.8976329719
30	0.1098596993	0.1098602368	0.0000005375	0.1098599681	0.8976329719
31	0.1098596993	0.1098600315	0.0000003322	0.1098598654	0.8976329719
32	0.1098598262	0.1098600315	0.0000002053	0.1098599288	0.8976329719
33	0.1098598262	0.1098599531	0.0000001269	0.1098598896	0.8976329719
34	0.1098598747	0.1098599531	0.0000000784	0.1098599139	0.8976329719
35	0.1098599046	0.1098599531	0.0000000485	0.1098599288	0.8976329719
36	0.1098599046	0.1098599346	0.0000000300	0.1098599196	0.8976329719
37	0.1098599046	0.1098599231	0.0000000185	0.1098599139	0.8976329719
38	0.1098599117	0.1098599231	0.0000000114	0.1098599174	0.8976329719
39	0.1098599117	0.1098599188	0.0000000071	0.1098599152	0.8976329719
40	0.1098599144	0.1098599188	0.0000000044	0.1098599166	0.8976329719
41	0.1098599161	0.1098599188	0.0000000027	0.1098599174	0.8976329719

Зависимость числа шагов от $\log \varepsilon$



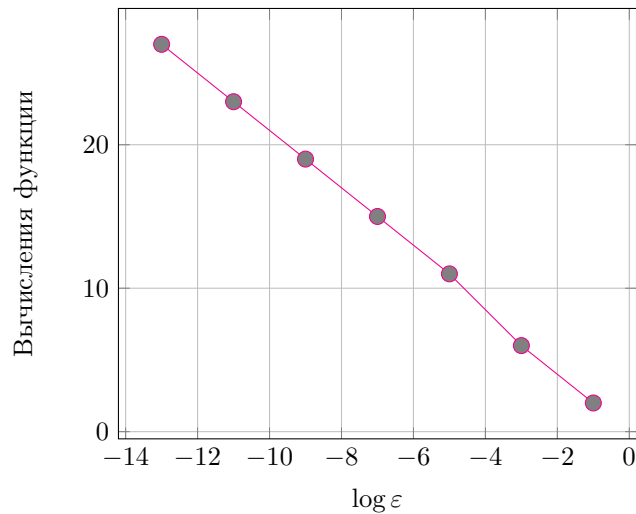
2.3.1 Вывод

По результатам нашего исследования, метод золотого сечения требует меньшего числа вычислений функции, чем метод дихотомии, потому что на каждом промежутке вычисляется только одно новое значение x и $f(x)$, потому его можно считать более эффективным. Однако, при малых ε он быстро теряет свое преимущество, так как константа $\phi = \frac{\sqrt{5}-1}{2}$ представляется в ЭВМ неточно, из-за чего быстро накапливается погрешность. С другой стороны, в методе золотого сечения не вычисляются значения в крайних точках, это удобно, например, для оптимизации функций с вертикальными асимптотами. Стоит отметить, что результат алгоритма с уменьшением количества итераций почти не меняется относительно ε . Количество итераций для метода золотого сечения также линейно зависит от логарифма ε .

2.4 Метод Фибоначчи

№	left	right	length	x	$f(x)$
0	-0.5000000000	0.5000000000	1.0000000000	0.0000000000	1.0000000000
1	-0.1180339888	0.5000000000	0.6180339888	0.1909830056	0.9397125964
2	-0.1180339888	0.2639320225	0.3819660112	0.0729490169	0.9080836327
3	0.0278640450	0.2639320225	0.2360679775	0.1458980337	0.9065669717
4	0.0278640450	0.1737620787	0.1458980337	0.1008130619	0.8982358981
5	0.0835921350	0.1737620787	0.0901699438	0.1286771069	0.9001344998
6	0.0835921350	0.1393202250	0.0557280900	0.1114561800	0.8976514481
7	0.0835921350	0.1180339888	0.0344418538	0.1008130619	0.8982358981
8	0.0967477525	0.1180339888	0.0212862362	0.1073908706	0.8976774447
9	0.1048783712	0.1180339888	0.0131556175	0.1114561800	0.8976514481
10	0.1048783712	0.1130089900	0.0081306187	0.1089436806	0.8976390820
11	0.1079839912	0.1130089900	0.0050249988	0.1104964906	0.8976359145
12	0.1079839912	0.1110896111	0.0031056199	0.1095368011	0.8976337311
13	0.1091702322	0.1110896111	0.0019193789	0.1101299216	0.8976335016
14	0.1091702322	0.1103564732	0.0011862410	0.1097633527	0.8976330397
15	0.1096233354	0.1103564732	0.0007331378	0.1099899043	0.8976330947
16	0.1096233354	0.1100764386	0.0004531032	0.1098498870	0.8976329726
17	0.1097964040	0.1100764386	0.0002800346	0.1099364213	0.8976330144
18	0.1097964040	0.1099694726	0.0001730686	0.1098829383	0.8976329757
19	0.1097964040	0.1099033700	0.0001069660	0.1098498870	0.8976329726
20	0.1098372674	0.1099033700	0.0000661026	0.1098703187	0.8976329727
21	0.1098372674	0.1098781309	0.0000408634	0.1098576991	0.8976329719
22	0.1098528917	0.1098781309	0.0000252392	0.1098655113	0.8976329721
23	0.1098528917	0.1098685159	0.0000156242	0.1098607038	0.8976329719
24	0.1098528917	0.1098625066	0.0000096149	0.1098576991	0.8976329719
25	0.1098564973	0.1098625066	0.0000060093	0.1098595019	0.8976329719
26	0.1098589010	0.1098625066	0.0000036056	0.1098607038	0.8976329719
27	0.1098589010	0.1098613047	0.0000024037	0.1098601029	0.8976329719

Зависимость числа шагов от $\log \varepsilon$



2.4.1 Вывод

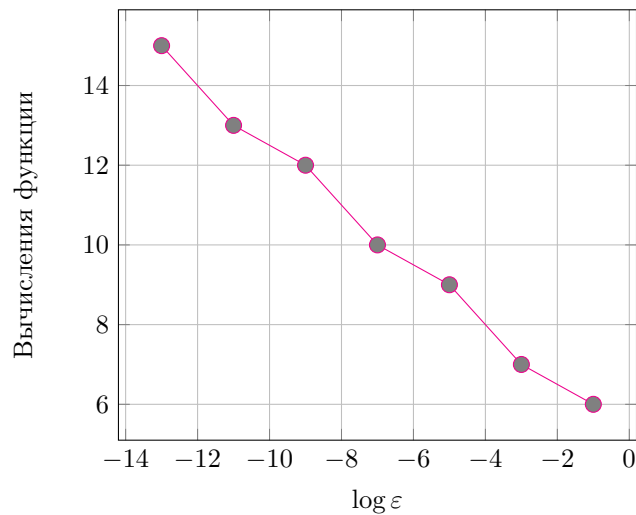
Количество итераций n фиксировано, его нужно выбирать исходя из точности и начальной длины промежутка. На нулевой итерации происходит два вычисления функции, на остальных - по одному. Как и в методе золотого сечения, в данном алгоритме не нужно считать значения в крайних точках. Однако метод Фибоначчи является более эффективным, чем метод золотого сечения, так как здесь коэффи-

циент сокращения интервала меняется от итерации к итерации. Как видно из графика, потребовалось меньше раз вычислять значение функции.

2.5 Метод парабол

№	left	right	length	x	$f(x)$
0	-0,5000000000	0,5000000000	1,0000000000	0,2213541667	0,9730619333
1	0,0000000000	0,5000000000	0,5000000000	0,1316301304	0,9009661307
2	0,0000000000	0,2213541667	0,2213541667	0,1193338695	0,8982761409
3	0,0000000000	0,1316301304	0,1316301304	0,1120409221	0,8976674337
4	0,0000000000	0,1193338695	0,1193338695	0,1106913723	0,8976379905
5	0,0000000000	0,1120409221	0,1120409221	0,1100749539	0,8976333079
6	0,0000000000	0,1106913723	0,1106913723	0,1099347383	0,8976330126
7	0,0000000000	0,1100749539	0,1100749539	0,1098806434	0,8976329750
8	0,0000000000	0,1099347383	0,1099347383	0,1098667490	0,8976329722
9	0,0000000000	0,1098806434	0,1098806434	0,1098618864	0,8976329719
10	0,0000000000	0,1098667490	0,1098667490	0,1098605449	0,8976329719
11	0,0000000000	0,1098618864	0,1098618864	0,1098601011	0,8976329719
12	0,0000000000	0,1098605449	0,1098605449	0,1098599734	0,8976329719
13	0,0000000000	0,1098601011	0,1098601011	0,1098599326	0,8976329719
14	0,0000000000	0,1098599734	0,1098599734	0,1098599205	0,8976329719
15	0,0000000000	0,1098599326	0,1098599326	0,1098599168	0,8976329719
16	0,0000000000	0,1098599205	0,1098599205	0,1098599151	0,8976329719

Зависимость числа шагов от $\log \varepsilon$



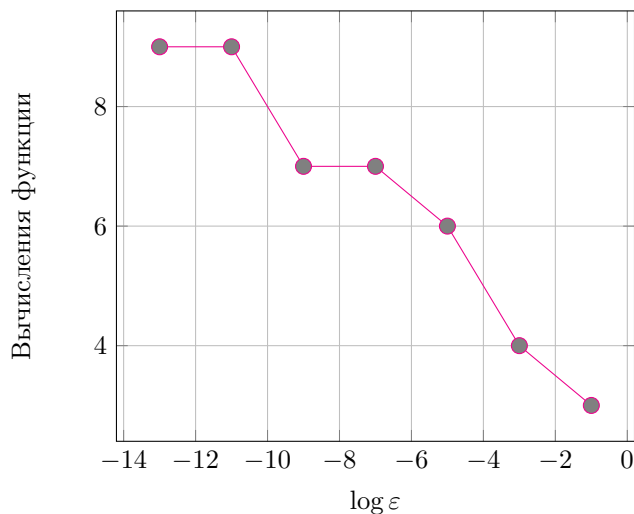
2.5.1 Вывод

Метод парабол требует вычисления значений в крайних точках функции, и нахождения промежуточной точки, удовлетворяющей неравенству на первой итерации. К тому же, в методе парабол меньше итераций по сравнению с предыдущими методами. В общем случае метод парабол обладает суперлинейной скоростью сходимости. Однако, такая высокая скорость сходимости гарантируется только в малой окрестности точки минимума. Количество итераций для метода парабол почти линейно зависит от логарифма ε .

2.6 Метод Брента

№	left	right	length	x	$f(x)$
0	-0.5000000000	0.5000000000	1.0000000000	0.0000000000	1.0000000000
1	-0.1180339887	0.5000000000	0.6180339887	0.1909830056	0.9397125964
2	-0.1180339887	0.2639320225	0.3819660113	0.0729490169	0.9080836327
3	-0.1180339887	0.1352549158	0.2532889046	0.0086104635	0.9835869750
4	-0.1180339887	0.1180339887	0.2360679775	-0.0000000000	1.0000000000
5	0.1043437684	0.1180339887	0.0136902203	0.1111888786	0.8976457835
6	0.1043437684	0.1104466173	0.0061028488	0.1073951929	0.8976772889
7	0.1043437684	0.1099089559	0.0055651875	0.1071263622	0.8976875053
8	0.1098426100	0.1099089559	0.0000663459	0.1098757830	0.8976329737
9	0.1098426100	0.1098618017	0.0000191917	0.1098522058	0.8976329723
10	0.1098426100	0.1098601107	0.0000175008	0.1098513604	0.8976329724
11	0.1098599018	0.1098601107	0.0000002089	0.1098600063	0.8976329719
12	0.1098599018	0.1098599250	0.0000000232	0.1098599134	0.8976329719
13	0.1098599154	0.1098599250	0.0000000096	0.1098599202	0.8976329719
14	0.1098599154	0.1098599198	0.0000000044	0.1098599176	0.8976329719
15	0.1098599154	0.1098599179	0.0000000025	0.1098599166	0.8976329719

Зависимость числа шагов от $\log \varepsilon$

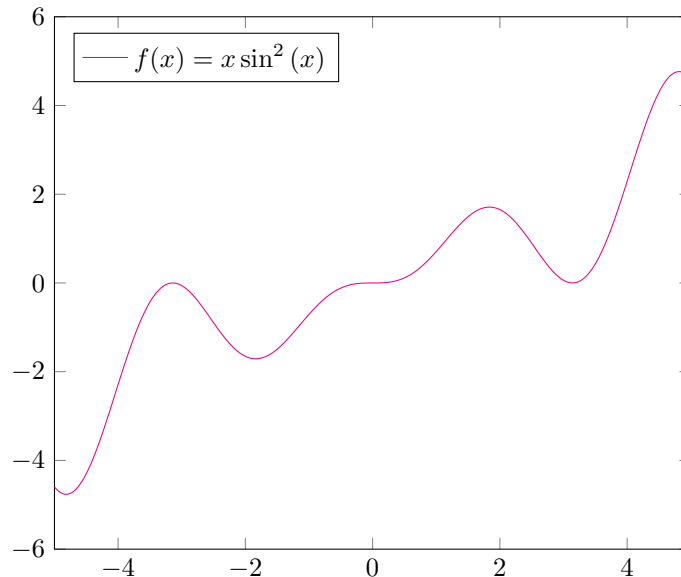


2.6.1 Вывод

Метод Брента эффективно комбинирует метод золотого сечения и метод парабол. На нашей функции в методе Брента было меньше всего вычислений функции. К его недостаткам можно отнести относительную сложность реализации по сравнению с остальными методами.

3 Тестирование алгоритмов на многомодальной функции

Приведённые алгоритмы работают при поиске минимума унимодальной функции. Оценим результаты их работы на многомодальной функции, например, вида $f(x) = x \sin^2(x)$ на отрезке $[-5; 5]$.



Для теста был выбран $\varepsilon = \exp^{-20}$. Аналитическим решением получили точку минимума: $x^* \approx -4.81584$. Ниже приведены результаты работы алгоритмов на данной функции.

метод	число итераций	результат
Дихотомия	34	-1.8365972455211614
Золотое сечение	47	-1.8365972013480927
Фибоначчи	28	-1.8366004038267394
Метод Парабол	∞	-
Метод Брента	∞	-

Как видно методы дихотомии, золотого сечения и Фибоначчи нашли точку локального минимума $x^* \approx -1.836$. Метод парабол, как и метод Брента, не завершился вообще. Проведённые тесты показывают, что данные методы не могут гарантировать корректный результат на многомодальных функциях.

4 Результаты исследования

В ходе работы нами были реализованы и протестированы пять алгоритмов поиска минимума унимодальной функции. Наиболее устойчивым оказался метод Брента, который выдаёт достаточно точный ответ вне зависимости от величины ε .

Меньше всего итераций и вычислений функции для нахождения ответа понадобилось методу Брента. Также мы протестировали алгоритмы на многомодальных функциях, и методы дихотомии, золотого сечения и Фибоначчи выдали точку локального минимума, а не глобального, метод парабол и Брента не завершились вообще. Мы не рекомендуем использовать данные методы на многомодальных функциях.