

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

Национальный исследовательский университет ИТМО

Мегафакультет трансляционных информационных технологий

Факультет информационных технологий и программирования

ЛАБОРАТОРНАЯ РАБОТА №1

По дисциплине «Прикладная математика»

Одномерная оптимизация нулевого порядка

Вариант 10

Выполнили Мельник Денис Александрович
Хакимов Руслан Венирович
Чечулин Лев Олегович

Проверила Гомозова Валерия Эдуардовна
(Фамилия, имя, отчество)

Санкт-Петербург, 2022 г.

ЗАДАНИЕ

1. Решить задачу в соответствии с номером варианта. Для решения реализовать алгоритмы одномерной минимизации функции без производной: метод дихотомии, метод золотого сечения, метод Фибоначчи, метод парабол и комбинированный метод Брента.

2. Сравните методы по количеству итераций и количеству вычислений функции в зависимости от разной точности. Для каждого метода обязательно указывайте, как изменяется отрезок при переходе к следующей итерации.

3. Протестировать реализованные алгоритмы для задач минимизации многомодальных функций, например, на различных полиномах. Могут ли метод золотого сечения/Брента не найти локальный минимум многомодальной функции?

Вариант 10. «Риски»

Предприимчивый бизнесмен затеял очередную авантюру, но решил подойти к этому вопросу системно и учесть все риски. Он исследовал риск от всех возможных предсказуемых факторов. Один из этих факторов, выраженный непрерывной величиной, дал наиболее интересную зависимость риска:

$$y(x) = \sin(0.5 * \ln(x) * x) + 1$$

Определите значение этого фактора, при котором авантюра бизнесмена будет наименее рискованной.

ХОД РАБОТЫ

Ссылка на репозиторий с реализацией: <https://github.com/primat-superteam/4th-semester/tree/main/lab1/lab1>

Для решения использовались следующие алгоритмы:

1. Метод дихотомии: на отрезке $[a; b]$ выбираются 2 точки: $\frac{a+b}{2} \pm \varepsilon$ и значения сравниваются в них. В результате происходит n итераций и $2n$ вычислений функции

Результаты для функции из условия:

Iteration #1	interval: [0.000010;0.500008]	interval length: 0.499998
Iteration #2	interval: [0.250006;0.500008]	interval length: 0.250003
Iteration #3	interval: [0.250006;0.375010]	interval length: 0.125005
Iteration #4	interval: [0.312505;0.375010]	interval length: 0.062506
Iteration #5	interval: [0.343754;0.375010]	interval length: 0.031256
Iteration #6	interval: [0.359379;0.375010]	interval length: 0.015631
Iteration #7	interval: [0.367191;0.375010]	interval length: 0.007819
Iteration #8	interval: [0.367191;0.371104]	interval length: 0.003913
Iteration #9	interval: [0.367191;0.369151]	interval length: 0.001960
Iteration #10	interval: [0.367191;0.368175]	interval length: 0.000983
Iteration #11	interval: [0.367680;0.368175]	interval length: 0.000495
Iteration #12	interval: [0.367680;0.367930]	interval length: 0.000251
Iteration #13	interval: [0.367802;0.367930]	interval length: 0.000129
Iteration #14	interval: [0.367863;0.367930]	interval length: 0.000068
Iteration #15	interval: [0.367863;0.367900]	interval length: 0.000037
Iteration #16	interval: [0.367863;0.367885]	interval length: 0.000022
Iteration #17	interval: [0.367870;0.367885]	interval length: 0.000014
Iteration #18	interval: [0.367874;0.367885]	interval length: 0.000010
Iteration #19	interval: [0.367876;0.367885]	interval length: 0.000009
Result:	0.36788	

2. Метод золотого сечения: на отрезке выбираются точки $a + (b - a) * \varphi$ и $b - (b - a) * \varphi$, где φ – золотое сечение. Благодаря этому при выборе точек для нового интервала одна из точек внутри уже удовлетворяет одному из этих соотношений. В итоге n итераций и $(n + 1)$ вычисление функции.

Результаты для функции из условия:

Iteration #1	interval: [0.000010;0.618038]	interval length: 0.618028
Iteration #2	interval: [0.236076;0.618038]	interval length: 0.381962
Iteration #3	interval: [0.236076;0.472141]	interval length: 0.236066
Iteration #4	interval: [0.326245;0.472141]	interval length: 0.145897
Iteration #5	interval: [0.326245;0.416414]	interval length: 0.090169
Iteration #6	interval: [0.326245;0.381972]	interval length: 0.055728
Iteration #7	interval: [0.347531;0.381972]	interval length: 0.034442
Iteration #8	interval: [0.360686;0.381972]	interval length: 0.021286
Iteration #9	interval: [0.360686;0.373842]	interval length: 0.013155
Iteration #10	interval: [0.365711;0.373842]	interval length: 0.008131
Iteration #11	interval: [0.365711;0.370736]	interval length: 0.005025

Одномерная оптимизация нулевого порядка

Iteration #12	interval: [0.365711;0.368817]	interval length: 0.003106
Iteration #13	interval: [0.366897;0.368817]	interval length: 0.001919
Iteration #14	interval: [0.367630;0.368817]	interval length: 0.001186
Iteration #15	interval: [0.367630;0.368364]	interval length: 0.000733
Iteration #16	interval: [0.367630;0.368084]	interval length: 0.000453
Iteration #17	interval: [0.367804;0.368084]	interval length: 0.000280
Iteration #18	interval: [0.367804;0.367977]	interval length: 0.000173
Iteration #19	interval: [0.367804;0.367911]	interval length: 0.000107
Iteration #20	interval: [0.367844;0.367911]	interval length: 0.000066
Iteration #21	interval: [0.367870;0.367911]	interval length: 0.000041
Iteration #22	interval: [0.367870;0.367895]	interval length: 0.000025
Iteration #23	interval: [0.367870;0.367885]	interval length: 0.000016
Iteration #24	interval: [0.367876;0.367885]	interval length: 0.000010
Result: 0.36788		

3. Метод Фибоначчи – если функцию желательно вычислить не более n раз, то можно выбирать точки $a + (b - a) * \frac{F_{n-2}}{F_n}$ и $a + (b - a) * \frac{F_{n-1}}{F_n}$ и затем пересчитывать их в $a + (b - a) * \frac{F_{n-i-1}}{F_{n-i}}$ или $a + (b - a) * \frac{F_{n-i-2}}{F_{n-i}}$. Метод работает, т.к. $\lim_{n \rightarrow \infty} \frac{F_{n+1}}{F_n} = \varphi$

Результаты для функции из условия (для $n = 20$):

Iteration #1	interval: [0.000010;0.618038]	interval length: 0.618028
Iteration #2	interval: [0.236076;0.618038]	interval length: 0.381962
Iteration #3	interval: [0.236076;0.472141]	interval length: 0.236066
Iteration #4	interval: [0.326245;0.472141]	interval length: 0.145897
Iteration #5	interval: [0.326245;0.416414]	interval length: 0.090169
Iteration #6	interval: [0.326245;0.381972]	interval length: 0.055727
Iteration #7	interval: [0.347531;0.381972]	interval length: 0.034442
Iteration #8	interval: [0.360686;0.381972]	interval length: 0.021286
Iteration #9	interval: [0.360686;0.373842]	interval length: 0.013156
Iteration #10	interval: [0.365712;0.373842]	interval length: 0.008130
Iteration #11	interval: [0.365712;0.370738]	interval length: 0.005026
Iteration #12	interval: [0.365712;0.368816]	interval length: 0.003104
Iteration #13	interval: [0.366895;0.368816]	interval length: 0.001922
Iteration #14	interval: [0.367634;0.368816]	interval length: 0.001183
Iteration #15	interval: [0.367634;0.368373]	interval length: 0.000739
Iteration #16	interval: [0.367634;0.368077]	interval length: 0.000443
Iteration #17	interval: [0.367782;0.368077]	interval length: 0.000296
Iteration #18	interval: [0.367782;0.367929]	interval length: 0.000148
Iteration #19	interval: [0.367782;0.367929]	interval length: 0.000148
Iteration #20	interval: [0.367782;0.367782]	interval length: 0.000000
Result: 0.367782		

4. Метод парабол – если есть предположение, что функция аппроксимируется квадратичной, то можно получить суперлинейную скорость сходимости. Выбирается 3 точки: $x_1 < x_2 < x_3$ и ищется вершина параболы u и происходит пересчёт точек из значений функции в точках x_2 и u . В итоге происходит n итераций и $(n + 3)$ вычислений значения функции

Результаты для функции из условия:

Iteration #1	interval: [0.000010;0.500005]	interval length: 0.499995
Iteration #2	interval: [0.000010;0.499963]	interval length: 0.499953

Одномерная оптимизация нулевого порядка

Iteration #3	interval: [0.000010;0.423801]	interval length: 0.423791
Iteration #4	interval: [0.000010;0.410126]	interval length: 0.410116
Iteration #5	interval: [0.000010;0.391112]	interval length: 0.391102
Iteration #6	interval: [0.000010;0.383734]	interval length: 0.383724
Iteration #7	interval: [0.000010;0.377504]	interval length: 0.377494
Iteration #8	interval: [0.000010;0.374212]	interval length: 0.374202
Iteration #9	interval: [0.000010;0.371870]	interval length: 0.371860
Iteration #10	interval: [0.000010;0.370471]	interval length: 0.370461
Iteration #11	interval: [0.000010;0.369536]	interval length: 0.369526
Iteration #12	interval: [0.000010;0.368950]	interval length: 0.368940
Iteration #13	interval: [0.000010;0.368567]	interval length: 0.368557
Iteration #14	interval: [0.000010;0.368323]	interval length: 0.368313
Iteration #15	interval: [0.000010;0.368165]	interval length: 0.368155
Iteration #16	interval: [0.000010;0.368064]	interval length: 0.368054
Iteration #17	interval: [0.000010;0.367998]	interval length: 0.367988
Iteration #18	interval: [0.000010;0.367956]	interval length: 0.367946
Iteration #19	interval: [0.000010;0.367929]	interval length: 0.367919
Iteration #20	interval: [0.000010;0.367911]	interval length: 0.367901
Result: 0.367896		

5. Метод Брента – комбинация метода золотого сечения и метода парабол. Метод золотого сечения сходится за гарантированное число итераций, а метод парабол обладает суперлинейной сходимостью в малой окрестности оптимального решения. Нужно отслеживать значения функции в 6-ти точках: a, b, x, w, v, u , где a, b – текущий интервал поиска решения, x – точка, соответствующая наименьшему из всех вычисленных на текущий момент значений, w – точка, в которой функция принимает второе снизу из всех вычисленных на текущий момент значений, v – предыдущее значение w , u – текущее приближение к точке минимума функции. По точкам x, w, v находится вершина параболы u . Если $u \notin [a, b]$ или $|u - x| > \frac{d_{prev}}{2}$, где d_{prev} – предыдущая длина интервала, то пересчитываем u как в методе золотого сечения. В итоге n итераций и n вычислений функции.

Результаты для функции из условия:

Iteration #1	interval: [0.190991;1.000000]	interval length: 0.809009
Iteration #2	interval: [0.190991;0.809019]	interval length: 0.618028
Iteration #3	interval: [0.190991;0.500005]	interval length: 0.309014
Iteration #4	interval: [0.190991;0.390986]	interval length: 0.199995
Iteration #5	interval: [0.190991;0.379774]	interval length: 0.188783
Iteration #6	interval: [0.365812;0.379774]	interval length: 0.013963
Iteration #7	interval: [0.367790;0.379774]	interval length: 0.011985
Iteration #8	interval: [0.367790;0.367891]	interval length: 0.000101
Iteration #9	interval: [0.367879;0.367891]	interval length: 0.000012
Iteration #10	interval: [0.367879;0.367879]	interval length: 0.000000
Result: 0.367879		

Были проведены тесты для многомодальных функций:

$x^3 + 10x - 1$:
Dixotomy:

Одномерная оптимизация нулевого порядка

Iteration #1	interval: [-1000.000000;0.000003]	interval length: 1000.000003
Iteration #2	interval: [-1000.000000;-499.999995]	interval length: 500.000005
Iteration #3	interval: [-1000.000000;-749.999994]	interval length: 250.000006
Iteration #4	interval: [-1000.000000;-874.999994]	interval length: 125.000006
Iteration #5	interval: [-1000.000000;-937.499994]	interval length: 62.500006
Iteration #6	interval: [-1000.000000;-968.749993]	interval length: 31.250007
Iteration #7	interval: [-1000.000000;-984.374993]	interval length: 15.625007
Iteration #8	interval: [-1000.000000;-992.187493]	interval length: 7.812507
Iteration #9	interval: [-1000.000000;-996.093743]	interval length: 3.906257
Iteration #10	interval: [-1000.000000;-998.046868]	interval length: 1.953132
Iteration #11	interval: [-1000.000000;-999.023431]	interval length: 0.976569
Iteration #12	interval: [-1000.000000;-999.511712]	interval length: 0.488288
Iteration #13	interval: [-1000.000000;-999.755853]	interval length: 0.244147
Iteration #14	interval: [-1000.000000;-999.877923]	interval length: 0.122077
Iteration #15	interval: [-1000.000000;-999.938958]	interval length: 0.061042
Iteration #16	interval: [-1000.000000;-999.969476]	interval length: 0.030524
Iteration #17	interval: [-1000.000000;-999.984735]	interval length: 0.015265
Iteration #18	interval: [-1000.000000;-999.992364]	interval length: 0.007636
Iteration #19	interval: [-1000.000000;-999.996179]	interval length: 0.003821
Iteration #20	interval: [-1000.000000;-999.998086]	interval length: 0.001914
Iteration #21	interval: [-1000.000000;-999.999040]	interval length: 0.000960
Iteration #22	interval: [-1000.000000;-999.999516]	interval length: 0.000484
Iteration #23	interval: [-1000.000000;-999.999755]	interval length: 0.000245
Iteration #24	interval: [-1000.000000;-999.999874]	interval length: 0.000126
Iteration #25	interval: [-1000.000000;-999.999934]	interval length: 0.000066
Iteration #26	interval: [-1000.000000;-999.999964]	interval length: 0.000036
Iteration #27	interval: [-1000.000000;-999.999978]	interval length: 0.000022
Iteration #28	interval: [-1000.000000;-999.999986]	interval length: 0.000014
Iteration #29	interval: [-1000.000000;-999.999990]	interval length: 0.000010
Iteration #30	interval: [-1000.000000;-999.999991]	interval length: 0.000009
Result:	-1000	

Golden section:

Iteration #1	interval: [-1000.000000;236.067977]	interval length: 1236.067977
Iteration #2	interval: [-1000.000000;-236.067977]	interval length: 763.932023
Iteration #3	interval: [-1000.000000;-527.864045]	interval length: 472.135955
Iteration #4	interval: [-1000.000000;-708.203932]	interval length: 291.796068
Iteration #5	interval: [-1000.000000;-819.660113]	interval length: 180.339887
Iteration #6	interval: [-1000.000000;-888.543820]	interval length: 111.456180
Iteration #7	interval: [-1000.000000;-931.116293]	interval length: 68.883707
Iteration #8	interval: [-1000.000000;-957.427527]	interval length: 42.572473
Iteration #9	interval: [-1000.000000;-973.688765]	interval length: 26.311235
Iteration #10	interval: [-1000.000000;-983.738762]	interval length: 16.261238
Iteration #11	interval: [-1000.000000;-989.950003]	interval length: 10.049997
Iteration #12	interval: [-1000.000000;-993.788760]	interval length: 6.211240
Iteration #13	interval: [-1000.000000;-996.161243]	interval length: 3.838757
Iteration #14	interval: [-1000.000000;-997.627517]	interval length: 2.372483
Iteration #15	interval: [-1000.000000;-998.533725]	interval length: 1.466275
Iteration #16	interval: [-1000.000000;-999.093792]	interval length: 0.906208
Iteration #17	interval: [-1000.000000;-999.439933]	interval length: 0.560067
Iteration #18	interval: [-1000.000000;-999.653859]	interval length: 0.346141
Iteration #19	interval: [-1000.000000;-999.786073]	interval length: 0.213927
Iteration #20	interval: [-1000.000000;-999.867786]	interval length: 0.132214
Iteration #21	interval: [-1000.000000;-999.918287]	interval length: 0.081713
Iteration #22	interval: [-1000.000000;-999.949499]	interval length: 0.050501
Iteration #23	interval: [-1000.000000;-999.968789]	interval length: 0.031211

Одномерная оптимизация нулевого порядка

Iteration #24	interval: [-1000.000000;-999.980710]	interval length: 0.019290
Iteration #25	interval: [-1000.000000;-999.988078]	interval length: 0.011922
Iteration #26	interval: [-1000.000000;-999.992632]	interval length: 0.007368
Iteration #27	interval: [-1000.000000;-999.995446]	interval length: 0.004554
Iteration #28	interval: [-1000.000000;-999.997186]	interval length: 0.002814
Iteration #29	interval: [-1000.000000;-999.998261]	interval length: 0.001739
Iteration #30	interval: [-1000.000000;-999.998925]	interval length: 0.001075
Iteration #31	interval: [-1000.000000;-999.999336]	interval length: 0.000664
Iteration #32	interval: [-1000.000000;-999.999589]	interval length: 0.000411
Iteration #33	interval: [-1000.000000;-999.999746]	interval length: 0.000254
Iteration #34	interval: [-1000.000000;-999.999843]	interval length: 0.000157
Iteration #35	interval: [-1000.000000;-999.999903]	interval length: 0.000097
Iteration #36	interval: [-1000.000000;-999.999940]	interval length: 0.000060
Iteration #37	interval: [-1000.000000;-999.999963]	interval length: 0.000037
Iteration #38	interval: [-1000.000000;-999.999977]	interval length: 0.000023
Iteration #39	interval: [-1000.000000;-999.999986]	interval length: 0.000014
Iteration #40	interval: [-1000.000000;-999.999991]	interval length: 0.000009
Result: -1000		

Fibonacci:

Iteration #1	interval: [-1000.000000;236.067977]	interval length: 1236.067977
Iteration #2	interval: [-1000.000000;-236.067977]	interval length: 763.932023
Iteration #3	interval: [-1000.000000;-527.864045]	interval length: 472.135955
Iteration #4	interval: [-1000.000000;-708.203932]	interval length: 291.796068
Iteration #5	interval: [-1000.000000;-819.660113]	interval length: 180.339887
Iteration #6	interval: [-1000.000000;-888.543820]	interval length: 111.456180
Iteration #7	interval: [-1000.000000;-931.116293]	interval length: 68.883707
Iteration #8	interval: [-1000.000000;-957.427527]	interval length: 42.572473
Iteration #9	interval: [-1000.000000;-973.688765]	interval length: 26.311235
Iteration #10	interval: [-1000.000000;-983.738762]	interval length: 16.261238
Iteration #11	interval: [-1000.000000;-989.950003]	interval length: 10.049997
Iteration #12	interval: [-1000.000000;-993.788760]	interval length: 6.211240
Iteration #13	interval: [-1000.000000;-996.161243]	interval length: 3.838757
Iteration #14	interval: [-1000.000000;-997.627517]	interval length: 2.372483
Iteration #15	interval: [-1000.000000;-998.533725]	interval length: 1.466275
Iteration #16	interval: [-1000.000000;-999.093792]	interval length: 0.906208
Iteration #17	interval: [-1000.000000;-999.439933]	interval length: 0.560067
Iteration #18	interval: [-1000.000000;-999.653859]	interval length: 0.346141
Iteration #19	interval: [-1000.000000;-999.786073]	interval length: 0.213927
Iteration #20	interval: [-1000.000000;-999.867786]	interval length: 0.132214
Iteration #21	interval: [-1000.000000;-999.918287]	interval length: 0.081713
Iteration #22	interval: [-1000.000000;-999.949499]	interval length: 0.050501
Iteration #23	interval: [-1000.000000;-999.968789]	interval length: 0.031211
Iteration #24	interval: [-1000.000000;-999.980710]	interval length: 0.019290
Iteration #25	interval: [-1000.000000;-999.988078]	interval length: 0.011922
Iteration #26	interval: [-1000.000000;-999.992632]	interval length: 0.007368
Iteration #27	interval: [-1000.000000;-999.995446]	interval length: 0.004554
Iteration #28	interval: [-1000.000000;-999.997186]	interval length: 0.002814
Iteration #29	interval: [-1000.000000;-999.998261]	interval length: 0.001739
Iteration #30	interval: [-1000.000000;-999.998925]	interval length: 0.001075
Iteration #31	interval: [-1000.000000;-999.999336]	interval length: 0.000664
Iteration #32	interval: [-1000.000000;-999.999589]	interval length: 0.000411
Iteration #33	interval: [-1000.000000;-999.999746]	interval length: 0.000254
Iteration #34	interval: [-1000.000000;-999.999843]	interval length: 0.000157
Iteration #35	interval: [-1000.000000;-999.999903]	interval length: 0.000097
Iteration #36	interval: [-1000.000000;-999.999940]	interval length: 0.000060

Одномерная оптимизация нулевого порядка

Iteration #37	interval: [-1000.000000;-999.999963]	interval length: 0.000037
Iteration #38	interval: [-1000.000000;-999.999977]	interval length: 0.000023
Iteration #39	interval: [-1000.000000;-999.999986]	interval length: 0.000014
Iteration #40	interval: [-1000.000000;-999.999991]	interval length: 0.000009
Iteration #41	interval: [-1000.000000;-999.999995]	interval length: 0.000005
Iteration #42	interval: [-1000.000000;-999.999997]	interval length: 0.000003
Iteration #43	interval: [-1000.000000;-999.999998]	interval length: 0.000002
Iteration #44	interval: [-1000.000000;-999.999999]	interval length: 0.000001
Iteration #45	interval: [-1000.000000;-999.999999]	interval length: 0.000001
Iteration #46	interval: [-1000.000000;-1000.000000]	interval length: 0.000000
Iteration #47	interval: [-1000.000000;-1000.000000]	interval length: 0.000000
Iteration #48	interval: [-1000.000000;-1000.000000]	interval length: 0.000000
Iteration #49	interval: [-1000.000000;-1000.000000]	interval length: 0.000000
Iteration #50	interval: [-1000.000000;-1000.000000]	interval length: 0.000000

Result: -1000

Parabolas:

Iteration #1	interval: [-1000.000000;inf]	interval length: inf
--------------	------------------------------	----------------------

Result: -nan(ind)

Brent:

Iteration #1	interval: [-1000.000000;0.000000]	interval length: 1000.000000
Iteration #2	interval: [-1000.000000;-236.067977]	interval length: 763.932023
Iteration #3	interval: [-1000.000000;-618.033989]	interval length: 381.966011
Iteration #4	interval: [-1000.000000;-708.203932]	interval length: 291.796068
Iteration #5	interval: [-1000.000000;-854.101966]	interval length: 145.898034
Iteration #6	interval: [-1000.000000;-888.543820]	interval length: 111.456180
Iteration #7	interval: [-1000.000000;-944.271910]	interval length: 55.728090
Iteration #8	interval: [-1000.000000;-957.427527]	interval length: 42.572473
Iteration #9	interval: [-1000.000000;-965.558146]	interval length: 34.441854
Iteration #10	interval: [-1000.000000;-978.713764]	interval length: 21.286236
Iteration #11	interval: [-1000.000000;-983.738762]	interval length: 16.261238
Iteration #12	interval: [-1000.000000;-991.869381]	interval length: 8.130619
Iteration #13	interval: [-1000.000000;-993.788760]	interval length: 6.211240
Iteration #14	interval: [-1000.000000;-996.894380]	interval length: 3.105620
Iteration #15	interval: [-1000.000000;-997.627517]	interval length: 2.372483
Iteration #16	interval: [-1000.000000;-998.813759]	interval length: 1.186241
Iteration #17	interval: [-1000.000000;-999.093792]	interval length: 0.906208
Iteration #18	interval: [-1000.000000;-999.546896]	interval length: 0.453104
Iteration #19	interval: [-1000.000000;-999.653859]	interval length: 0.346141
Iteration #20	interval: [-1000.000000;-999.826930]	interval length: 0.173070
Iteration #21	interval: [-1000.000000;-999.867786]	interval length: 0.132214
Iteration #22	interval: [-1000.000000;-999.933893]	interval length: 0.066107
Iteration #23	interval: [-1000.000000;-999.949499]	interval length: 0.050501
Iteration #24	interval: [-1000.000000;-999.974749]	interval length: 0.025251
Iteration #25	interval: [-1000.000000;-999.980710]	interval length: 0.019290
Iteration #26	interval: [-1000.000000;-999.984394]	interval length: 0.015606
Iteration #27	interval: [-1000.000000;-999.990355]	interval length: 0.009645
Iteration #28	interval: [-1000.000000;-999.992632]	interval length: 0.007368
Iteration #29	interval: [-1000.000000;-999.996316]	interval length: 0.003684
Iteration #30	interval: [-1000.000000;-999.997186]	interval length: 0.002814
Iteration #31	interval: [-1000.000000;-999.998593]	interval length: 0.001407
Iteration #32	interval: [-1000.000000;-999.998925]	interval length: 0.001075
Iteration #33	interval: [-1000.000000;-999.999463]	interval length: 0.000537
Iteration #34	interval: [-1000.000000;-999.999589]	interval length: 0.000411
Iteration #35	interval: [-1000.000000;-999.999795]	interval length: 0.000205

Одномерная оптимизация нулевого порядка

Iteration #36	interval: [-1000.000000;-999.999843]	interval length: 0.000157
Iteration #37	interval: [-1000.000000;-999.999922]	interval length: 0.000078
Iteration #38	interval: [-1000.000000;-999.999940]	interval length: 0.000060
Iteration #39	interval: [-1000.000000;-999.999970]	interval length: 0.000030
Iteration #40	interval: [-1000.000000;-999.999977]	interval length: 0.000023
Iteration #41	interval: [-1000.000000;-999.999989]	interval length: 0.000011
Iteration #42	interval: [-1000.000000;-999.999991]	interval length: 0.000009

Result: -1000

$x^5 + 3x^3 - 6x + 12$:

Dixotomy:

Iteration #1	interval: [-0.000003;100.000000]	interval length: 100.000003
Iteration #2	interval: [-0.000003;50.000002]	interval length: 50.000005
Iteration #3	interval: [-0.000003;25.000002]	interval length: 25.000006
Iteration #4	interval: [-0.000003;12.500003]	interval length: 12.500006
Iteration #5	interval: [-0.000003;6.250003]	interval length: 6.250006
Iteration #6	interval: [-0.000003;3.125003]	interval length: 3.125007
Iteration #7	interval: [-0.000003;1.562503]	interval length: 1.562507
Iteration #8	interval: [-0.000003;0.781253]	interval length: 0.781257
Iteration #9	interval: [0.390622;0.781253]	interval length: 0.390632
Iteration #10	interval: [0.585934;0.781253]	interval length: 0.195319
Iteration #11	interval: [0.683590;0.781253]	interval length: 0.097663
Iteration #12	interval: [0.683590;0.732425]	interval length: 0.048835
Iteration #13	interval: [0.708004;0.732425]	interval length: 0.024421
Iteration #14	interval: [0.708004;0.720218]	interval length: 0.012214
Iteration #15	interval: [0.714108;0.720218]	interval length: 0.006110
Iteration #16	interval: [0.717160;0.720218]	interval length: 0.003058
Iteration #17	interval: [0.718686;0.720218]	interval length: 0.001533
Iteration #18	interval: [0.719449;0.720218]	interval length: 0.000770
Iteration #19	interval: [0.719449;0.719837]	interval length: 0.000388
Iteration #20	interval: [0.719449;0.719646]	interval length: 0.000197
Iteration #21	interval: [0.719449;0.719551]	interval length: 0.000102
Iteration #22	interval: [0.719496;0.719551]	interval length: 0.000054
Iteration #23	interval: [0.719520;0.719551]	interval length: 0.000031
Iteration #24	interval: [0.719532;0.719551]	interval length: 0.000019
Iteration #25	interval: [0.719538;0.719551]	interval length: 0.000013
Iteration #26	interval: [0.719541;0.719551]	interval length: 0.000010

Result: 0.719546

Golden section:

Iteration #1	interval: [-100.000000;23.606798]	interval length: 123.606798
Iteration #2	interval: [-100.000000;-23.606798]	interval length: 76.393202
Iteration #3	interval: [-100.000000;-52.786405]	interval length: 47.213595
Iteration #4	interval: [-100.000000;-70.820393]	interval length: 29.179607
Iteration #5	interval: [-100.000000;-81.966011]	interval length: 18.033989
Iteration #6	interval: [-100.000000;-88.854382]	interval length: 11.145618
Iteration #7	interval: [-100.000000;-93.111629]	interval length: 6.888371
Iteration #8	interval: [-100.000000;-95.742753]	interval length: 4.257247
Iteration #9	interval: [-100.000000;-97.368877]	interval length: 2.631123
Iteration #10	interval: [-100.000000;-98.373876]	interval length: 1.626124
Iteration #11	interval: [-100.000000;-98.995000]	interval length: 1.005000
Iteration #12	interval: [-100.000000;-99.378876]	interval length: 0.621124
Iteration #13	interval: [-100.000000;-99.616124]	interval length: 0.383876
Iteration #14	interval: [-100.000000;-99.762752]	interval length: 0.237248
Iteration #15	interval: [-100.000000;-99.853373]	interval length: 0.146627
Iteration #16	interval: [-100.000000;-99.909379]	interval length: 0.090621

Одномерная оптимизация нулевого порядка

Iteration #17	interval: [-100.000000;-99.943993]	interval length: 0.056007
Iteration #18	interval: [-100.000000;-99.965386]	interval length: 0.034614
Iteration #19	interval: [-100.000000;-99.978607]	interval length: 0.021393
Iteration #20	interval: [-100.000000;-99.986779]	interval length: 0.013221
Iteration #21	interval: [-100.000000;-99.991829]	interval length: 0.008171
Iteration #22	interval: [-100.000000;-99.994950]	interval length: 0.005050
Iteration #23	interval: [-100.000000;-99.996879]	interval length: 0.003121
Iteration #24	interval: [-100.000000;-99.998071]	interval length: 0.001929
Iteration #25	interval: [-100.000000;-99.998808]	interval length: 0.001192
Iteration #26	interval: [-100.000000;-99.999263]	interval length: 0.000737
Iteration #27	interval: [-100.000000;-99.999545]	interval length: 0.000455
Iteration #28	interval: [-100.000000;-99.999719]	interval length: 0.000281
Iteration #29	interval: [-100.000000;-99.999826]	interval length: 0.000174
Iteration #30	interval: [-100.000000;-99.999893]	interval length: 0.000107
Iteration #31	interval: [-100.000000;-99.999934]	interval length: 0.000066
Iteration #32	interval: [-100.000000;-99.999959]	interval length: 0.000041
Iteration #33	interval: [-100.000000;-99.999975]	interval length: 0.000025
Iteration #34	interval: [-100.000000;-99.999984]	interval length: 0.000016
Iteration #35	interval: [-100.000000;-99.999990]	interval length: 0.000010

Result: -100

Fibonacci:

Iteration #1	interval: [-100.000000;23.606798]	interval length: 123.606798
Iteration #2	interval: [-100.000000;-23.606798]	interval length: 76.393202
Iteration #3	interval: [-100.000000;-52.786405]	interval length: 47.213595
Iteration #4	interval: [-100.000000;-70.820393]	interval length: 29.179607
Iteration #5	interval: [-100.000000;-81.966011]	interval length: 18.033989
Iteration #6	interval: [-100.000000;-88.854382]	interval length: 11.145618
Iteration #7	interval: [-100.000000;-93.111629]	interval length: 6.888371
Iteration #8	interval: [-100.000000;-95.742753]	interval length: 4.257247
Iteration #9	interval: [-100.000000;-97.368877]	interval length: 2.631123
Iteration #10	interval: [-100.000000;-98.373876]	interval length: 1.626124
Iteration #11	interval: [-100.000000;-98.995000]	interval length: 1.005000
Iteration #12	interval: [-100.000000;-99.378876]	interval length: 0.621124
Iteration #13	interval: [-100.000000;-99.616124]	interval length: 0.383876
Iteration #14	interval: [-100.000000;-99.762752]	interval length: 0.237248
Iteration #15	interval: [-100.000000;-99.853373]	interval length: 0.146627
Iteration #16	interval: [-100.000000;-99.909379]	interval length: 0.090621
Iteration #17	interval: [-100.000000;-99.943993]	interval length: 0.056007
Iteration #18	interval: [-100.000000;-99.965386]	interval length: 0.034614
Iteration #19	interval: [-100.000000;-99.978607]	interval length: 0.021393
Iteration #20	interval: [-100.000000;-99.986779]	interval length: 0.013221
Iteration #21	interval: [-100.000000;-99.991829]	interval length: 0.008171
Iteration #22	interval: [-100.000000;-99.994950]	interval length: 0.005050
Iteration #23	interval: [-100.000000;-99.996879]	interval length: 0.003121
Iteration #24	interval: [-100.000000;-99.998071]	interval length: 0.001929
Iteration #25	interval: [-100.000000;-99.998808]	interval length: 0.001192
Iteration #26	interval: [-100.000000;-99.999263]	interval length: 0.000737
Iteration #27	interval: [-100.000000;-99.999545]	interval length: 0.000455
Iteration #28	interval: [-100.000000;-99.999719]	interval length: 0.000281
Iteration #29	interval: [-100.000000;-99.999826]	interval length: 0.000174
Iteration #30	interval: [-100.000000;-99.999893]	interval length: 0.000107
Iteration #31	interval: [-100.000000;-99.999934]	interval length: 0.000066
Iteration #32	interval: [-100.000000;-99.999959]	interval length: 0.000041
Iteration #33	interval: [-100.000000;-99.999975]	interval length: 0.000025
Iteration #34	interval: [-100.000000;-99.999984]	interval length: 0.000016

Одномерная оптимизация нулевого порядка

Iteration #35	interval: [-100.000000;-99.999990]	interval length: 0.000010
Iteration #36	interval: [-100.000000;-99.999994]	interval length: 0.000006
Iteration #37	interval: [-100.000000;-99.999996]	interval length: 0.000004
Iteration #38	interval: [-100.000000;-99.999998]	interval length: 0.000002
Iteration #39	interval: [-100.000000;-99.999999]	interval length: 0.000001
Iteration #40	interval: [-100.000000;-99.999999]	interval length: 0.000001
Iteration #41	interval: [-100.000000;-99.999999]	interval length: 0.000001
Iteration #42	interval: [-100.000000;-100.000000]	interval length: 0.000000
Iteration #43	interval: [-100.000000;-100.000000]	interval length: 0.000000
Iteration #44	interval: [-100.000000;-100.000000]	interval length: 0.000000
Iteration #45	interval: [-100.000000;-100.000000]	interval length: 0.000000
Iteration #46	interval: [-100.000000;-100.000000]	interval length: 0.000000
Iteration #47	interval: [-100.000000;-100.000000]	interval length: 0.000000
Iteration #48	interval: [-100.000000;-100.000000]	interval length: 0.000000
Iteration #49	interval: [-100.000000;-100.000000]	interval length: 0.000000
Iteration #50	interval: [-100.000000;-100.000000]	interval length: 0.000000

Result: -100

Parabolas:

Iteration #1	interval: [-100.000000;inf]	interval length: inf
--------------	-----------------------------	----------------------

Result: -nan(ind)

Brent:

Iteration #1	interval: [-100.000000;0.000000]	interval length: 100.000000
Iteration #2	interval: [-100.000000;-23.606798]	interval length: 76.393202
Iteration #3	interval: [-100.000000;-38.196601]	interval length: 61.803399
Iteration #4	interval: [-100.000000;-61.803399]	interval length: 38.196601
Iteration #5	interval: [-100.000000;-70.820393]	interval length: 29.179607
Iteration #6	interval: [-100.000000;-76.393202]	interval length: 23.606798
Iteration #7	interval: [-100.000000;-85.410197]	interval length: 14.589803
Iteration #8	interval: [-100.000000;-88.854382]	interval length: 11.145618
Iteration #9	interval: [-100.000000;-94.427191]	interval length: 5.572809
Iteration #10	interval: [-100.000000;-95.742753]	interval length: 4.257247
Iteration #11	interval: [-100.000000;-96.555815]	interval length: 3.444185
Iteration #12	interval: [-100.000000;-97.871376]	interval length: 2.128624
Iteration #13	interval: [-100.000000;-98.373876]	interval length: 1.626124
Iteration #14	interval: [-100.000000;-98.684438]	interval length: 1.315562
Iteration #15	interval: [-100.000000;-99.186938]	interval length: 0.813062
Iteration #16	interval: [-100.000000;-99.378876]	interval length: 0.621124
Iteration #17	interval: [-100.000000;-99.689438]	interval length: 0.310562
Iteration #18	interval: [-100.000000;-99.762752]	interval length: 0.237248
Iteration #19	interval: [-100.000000;-99.881376]	interval length: 0.118624
Iteration #20	interval: [-100.000000;-99.909379]	interval length: 0.090621
Iteration #21	interval: [-100.000000;-99.954690]	interval length: 0.045310
Iteration #22	interval: [-100.000000;-99.965386]	interval length: 0.034614
Iteration #23	interval: [-100.000000;-99.982693]	interval length: 0.017307
Iteration #24	interval: [-100.000000;-99.986779]	interval length: 0.013221
Iteration #25	interval: [-100.000000;-99.993389]	interval length: 0.006611
Iteration #26	interval: [-100.000000;-99.994950]	interval length: 0.005050
Iteration #27	interval: [-100.000000;-99.997475]	interval length: 0.002525
Iteration #28	interval: [-100.000000;-99.998071]	interval length: 0.001929
Iteration #29	interval: [-100.000000;-99.999036]	interval length: 0.000964
Iteration #30	interval: [-100.000000;-99.999263]	interval length: 0.000737
Iteration #31	interval: [-100.000000;-99.999632]	interval length: 0.000368
Iteration #32	interval: [-100.000000;-99.999719]	interval length: 0.000281
Iteration #33	interval: [-100.000000;-99.999859]	interval length: 0.000141

Одномерная оптимизация нулевого порядка

Iteration #34	interval: [-100.000000;-99.999893]	interval length: 0.000107
Iteration #35	interval: [-100.000000;-99.999946]	interval length: 0.000054
Iteration #36	interval: [-100.000000;-99.999959]	interval length: 0.000041
Iteration #37	interval: [-100.000000;-99.999979]	interval length: 0.000021
Iteration #38	interval: [-100.000000;-99.999984]	interval length: 0.000016
Iteration #39	interval: [-100.000000;-99.999992]	interval length: 0.000008

Result: -100

$x^7 + x^5 - x^3 - 10x + 100$:

Dixotomy:

Iteration #1	interval: [-0.000003;10.000000]	interval length: 10.000003
Iteration #2	interval: [-0.000003;5.000002]	interval length: 5.000005
Iteration #3	interval: [-0.000003;2.500002]	interval length: 2.500006
Iteration #4	interval: [-0.000003;1.250003]	interval length: 1.250006
Iteration #5	interval: [0.624996;1.250003]	interval length: 0.625006
Iteration #6	interval: [0.937496;1.250003]	interval length: 0.312507
Iteration #7	interval: [0.937496;1.093753]	interval length: 0.156257
Iteration #8	interval: [1.015621;1.093753]	interval length: 0.078132
Iteration #9	interval: [1.015621;1.054690]	interval length: 0.039069
Iteration #10	interval: [1.015621;1.035159]	interval length: 0.019538
Iteration #11	interval: [1.015621;1.025394]	interval length: 0.009772
Iteration #12	interval: [1.015621;1.020511]	interval length: 0.004889
Iteration #13	interval: [1.015621;1.018069]	interval length: 0.002448
Iteration #14	interval: [1.016842;1.018069]	interval length: 0.001227
Iteration #15	interval: [1.016842;1.017459]	interval length: 0.000617
Iteration #16	interval: [1.016842;1.017154]	interval length: 0.000312
Iteration #17	interval: [1.016995;1.017154]	interval length: 0.000159
Iteration #18	interval: [1.017071;1.017154]	interval length: 0.000083
Iteration #19	interval: [1.017109;1.017154]	interval length: 0.000045
Iteration #20	interval: [1.017128;1.017154]	interval length: 0.000026
Iteration #21	interval: [1.017138;1.017154]	interval length: 0.000016
Iteration #22	interval: [1.017142;1.017154]	interval length: 0.000011
Iteration #23	interval: [1.017145;1.017154]	interval length: 0.000009

Result: 1.01715

Golden section:

Iteration #1	interval: [-10.000000;2.360680]	interval length: 12.360680
Iteration #2	interval: [-10.000000;-2.360680]	interval length: 7.639320
Iteration #3	interval: [-10.000000;-5.278640]	interval length: 4.721360
Iteration #4	interval: [-10.000000;-7.082039]	interval length: 2.917961
Iteration #5	interval: [-10.000000;-8.196601]	interval length: 1.803399
Iteration #6	interval: [-10.000000;-8.885438]	interval length: 1.114562
Iteration #7	interval: [-10.000000;-9.311163]	interval length: 0.688837
Iteration #8	interval: [-10.000000;-9.574275]	interval length: 0.425725
Iteration #9	interval: [-10.000000;-9.736888]	interval length: 0.263112
Iteration #10	interval: [-10.000000;-9.837388]	interval length: 0.162612
Iteration #11	interval: [-10.000000;-9.899500]	interval length: 0.100500
Iteration #12	interval: [-10.000000;-9.937888]	interval length: 0.062112
Iteration #13	interval: [-10.000000;-9.961612]	interval length: 0.038388
Iteration #14	interval: [-10.000000;-9.976275]	interval length: 0.023725
Iteration #15	interval: [-10.000000;-9.985337]	interval length: 0.014663
Iteration #16	interval: [-10.000000;-9.990938]	interval length: 0.009062
Iteration #17	interval: [-10.000000;-9.994399]	interval length: 0.005601
Iteration #18	interval: [-10.000000;-9.996539]	interval length: 0.003461
Iteration #19	interval: [-10.000000;-9.997861]	interval length: 0.002139
Iteration #20	interval: [-10.000000;-9.998678]	interval length: 0.001322

Одномерная оптимизация нулевого порядка

Iteration #21	interval: [-10.000000;-9.999183]	interval length: 0.000817
Iteration #22	interval: [-10.000000;-9.999495]	interval length: 0.000505
Iteration #23	interval: [-10.000000;-9.999688]	interval length: 0.000312
Iteration #24	interval: [-10.000000;-9.999807]	interval length: 0.000193
Iteration #25	interval: [-10.000000;-9.999881]	interval length: 0.000119
Iteration #26	interval: [-10.000000;-9.999926]	interval length: 0.000074
Iteration #27	interval: [-10.000000;-9.999954]	interval length: 0.000046
Iteration #28	interval: [-10.000000;-9.999972]	interval length: 0.000028
Iteration #29	interval: [-10.000000;-9.999983]	interval length: 0.000017
Iteration #30	interval: [-10.000000;-9.999989]	interval length: 0.000011
Iteration #31	interval: [-10.000000;-9.999993]	interval length: 0.000007
Result: -10		

Fibonacci:

Iteration #1	interval: [-10.000000;2.360680]	interval length: 12.360680
Iteration #2	interval: [-10.000000;-2.360680]	interval length: 7.639320
Iteration #3	interval: [-10.000000;-5.278640]	interval length: 4.721360
Iteration #4	interval: [-10.000000;-7.082039]	interval length: 2.917961
Iteration #5	interval: [-10.000000;-8.196601]	interval length: 1.803399
Iteration #6	interval: [-10.000000;-8.885438]	interval length: 1.114562
Iteration #7	interval: [-10.000000;-9.311163]	interval length: 0.688837
Iteration #8	interval: [-10.000000;-9.574275]	interval length: 0.425725
Iteration #9	interval: [-10.000000;-9.736888]	interval length: 0.263112
Iteration #10	interval: [-10.000000;-9.837388]	interval length: 0.162612
Iteration #11	interval: [-10.000000;-9.899500]	interval length: 0.100500
Iteration #12	interval: [-10.000000;-9.937888]	interval length: 0.062112
Iteration #13	interval: [-10.000000;-9.961612]	interval length: 0.038388
Iteration #14	interval: [-10.000000;-9.976275]	interval length: 0.023725
Iteration #15	interval: [-10.000000;-9.985337]	interval length: 0.014663
Iteration #16	interval: [-10.000000;-9.990938]	interval length: 0.009062
Iteration #17	interval: [-10.000000;-9.994399]	interval length: 0.005601
Iteration #18	interval: [-10.000000;-9.996539]	interval length: 0.003461
Iteration #19	interval: [-10.000000;-9.997861]	interval length: 0.002139
Iteration #20	interval: [-10.000000;-9.998678]	interval length: 0.001322
Iteration #21	interval: [-10.000000;-9.999183]	interval length: 0.000817
Iteration #22	interval: [-10.000000;-9.999495]	interval length: 0.000505
Iteration #23	interval: [-10.000000;-9.999688]	interval length: 0.000312
Iteration #24	interval: [-10.000000;-9.999807]	interval length: 0.000193
Iteration #25	interval: [-10.000000;-9.999881]	interval length: 0.000119
Iteration #26	interval: [-10.000000;-9.999926]	interval length: 0.000074
Iteration #27	interval: [-10.000000;-9.999954]	interval length: 0.000046
Iteration #28	interval: [-10.000000;-9.999972]	interval length: 0.000028
Iteration #29	interval: [-10.000000;-9.999983]	interval length: 0.000017
Iteration #30	interval: [-10.000000;-9.999989]	interval length: 0.000011
Iteration #31	interval: [-10.000000;-9.999993]	interval length: 0.000007
Iteration #32	interval: [-10.000000;-9.999996]	interval length: 0.000004
Iteration #33	interval: [-10.000000;-9.999997]	interval length: 0.000003
Iteration #34	interval: [-10.000000;-9.999998]	interval length: 0.000002
Iteration #35	interval: [-10.000000;-9.999999]	interval length: 0.000001
Iteration #36	interval: [-10.000000;-9.999999]	interval length: 0.000001
Iteration #37	interval: [-10.000000;-10.000000]	interval length: 0.000000
Iteration #38	interval: [-10.000000;-10.000000]	interval length: 0.000000
Iteration #39	interval: [-10.000000;-10.000000]	interval length: 0.000000
Iteration #40	interval: [-10.000000;-10.000000]	interval length: 0.000000
Iteration #41	interval: [-10.000000;-10.000000]	interval length: 0.000000
Iteration #42	interval: [-10.000000;-10.000000]	interval length: 0.000000

Одномерная оптимизация нулевого порядка

```
Iteration #43 interval: [-10.000000;-10.000000] interval length: 0.000000
Iteration #44 interval: [-10.000000;-10.000000] interval length: 0.000000
Iteration #45 interval: [-10.000000;-10.000000] interval length: 0.000000
Iteration #46 interval: [-10.000000;-10.000000] interval length: 0.000000
Iteration #47 interval: [-10.000000;-10.000000] interval length: 0.000000
Iteration #48 interval: [-10.000000;-10.000000] interval length: 0.000000
Iteration #49 interval: [-10.000000;-10.000000] interval length: 0.000000
Iteration #50 interval: [-10.000000;-10.000000] interval length: 0.000000
Result: -10
```

Parabolas:

```
Iteration #1 interval: [-10.000000;inf] interval length: inf
Result: -nan(ind)
```

Brent:

```
Iteration #1 interval: [-10.000000;0.000000] interval length: 10.000000
Iteration #2 interval: [-10.000000;-2.360680] interval length: 7.639320
Iteration #3 interval: [-10.000000;-3.819660] interval length: 6.180340
Iteration #4 interval: [-10.000000;-6.180340] interval length: 3.819660
Iteration #5 interval: [-10.000000;-7.082039] interval length: 2.917961
Iteration #6 interval: [-10.000000;-8.541020] interval length: 1.458980
Iteration #7 interval: [-10.000000;-8.885438] interval length: 1.114562
Iteration #8 interval: [-10.000000;-9.442719] interval length: 0.557281
Iteration #9 interval: [-10.000000;-9.574275] interval length: 0.425725
Iteration #10 interval: [-10.000000;-9.787138] interval length: 0.212862
Iteration #11 interval: [-10.000000;-9.837388] interval length: 0.162612
Iteration #12 interval: [-10.000000;-9.918694] interval length: 0.081306
Iteration #13 interval: [-10.000000;-9.937888] interval length: 0.062112
Iteration #14 interval: [-10.000000;-9.968944] interval length: 0.031056
Iteration #15 interval: [-10.000000;-9.976275] interval length: 0.023725
Iteration #16 interval: [-10.000000;-9.988138] interval length: 0.011862
Iteration #17 interval: [-10.000000;-9.990938] interval length: 0.009062
Iteration #18 interval: [-10.000000;-9.995469] interval length: 0.004531
Iteration #19 interval: [-10.000000;-9.996539] interval length: 0.003461
Iteration #20 interval: [-10.000000;-9.998269] interval length: 0.001731
Iteration #21 interval: [-10.000000;-9.998678] interval length: 0.001322
Iteration #22 interval: [-10.000000;-9.999339] interval length: 0.000661
Iteration #23 interval: [-10.000000;-9.999495] interval length: 0.000505
Iteration #24 interval: [-10.000000;-9.999747] interval length: 0.000253
Iteration #25 interval: [-10.000000;-9.999807] interval length: 0.000193
Iteration #26 interval: [-10.000000;-9.999844] interval length: 0.000156
Iteration #27 interval: [-10.000000;-9.999904] interval length: 0.000096
Iteration #28 interval: [-10.000000;-9.999926] interval length: 0.000074
Iteration #29 interval: [-10.000000;-9.999963] interval length: 0.000037
Iteration #30 interval: [-10.000000;-9.999972] interval length: 0.000028
Iteration #31 interval: [-10.000000;-9.999986] interval length: 0.000014
Iteration #32 interval: [-10.000000;-9.999989] interval length: 0.000011
Iteration #33 interval: [-10.000000;-9.999995] interval length: 0.000005
Result: -10
```

Видно, что методы находят один из минимумов или наименьшее значение функции. Более того, например, метод парабол может вообще не сходиться.

Пример, в котором методы золотого сечения и Брента не находят минимум:

Одномерная оптимизация нулевого порядка

функция $f(x) = x^3 - 6x + 1$, отрезок $[-15; 5]$

Результат:

Golden section:

Iteration #1	interval: [-15.000000; -2.639320]	interval length: 12.360680
Iteration #2	interval: [-15.000000; -7.360680]	interval length: 7.639320
Iteration #3	interval: [-15.000000; -10.278640]	interval length: 4.721360
Iteration #4	interval: [-15.000000; -12.082039]	interval length: 2.917961
Iteration #5	interval: [-15.000000; -13.196601]	interval length: 1.803399
Iteration #6	interval: [-15.000000; -13.885438]	interval length: 1.114562
Iteration #7	interval: [-15.000000; -14.311163]	interval length: 0.688837
Iteration #8	interval: [-15.000000; -14.574275]	interval length: 0.425725
Iteration #9	interval: [-15.000000; -14.736888]	interval length: 0.263112
Iteration #10	interval: [-15.000000; -14.837388]	interval length: 0.162612
Iteration #11	interval: [-15.000000; -14.899500]	interval length: 0.100500
Iteration #12	interval: [-15.000000; -14.937888]	interval length: 0.062112
Iteration #13	interval: [-15.000000; -14.961612]	interval length: 0.038388
Iteration #14	interval: [-15.000000; -14.976275]	interval length: 0.023725
Iteration #15	interval: [-15.000000; -14.985337]	interval length: 0.014663
Iteration #16	interval: [-15.000000; -14.990938]	interval length: 0.009062
Iteration #17	interval: [-15.000000; -14.994399]	interval length: 0.005601
Iteration #18	interval: [-15.000000; -14.996539]	interval length: 0.003461
Iteration #19	interval: [-15.000000; -14.997861]	interval length: 0.002139
Iteration #20	interval: [-15.000000; -14.998678]	interval length: 0.001322
Iteration #21	interval: [-15.000000; -14.999183]	interval length: 0.000817
Iteration #22	interval: [-15.000000; -14.999495]	interval length: 0.000505
Iteration #23	interval: [-15.000000; -14.999688]	interval length: 0.000312
Iteration #24	interval: [-15.000000; -14.999807]	interval length: 0.000193
Iteration #25	interval: [-15.000000; -14.999881]	interval length: 0.000119
Iteration #26	interval: [-15.000000; -14.999926]	interval length: 0.000074
Iteration #27	interval: [-15.000000; -14.999954]	interval length: 0.000046
Iteration #28	interval: [-15.000000; -14.999972]	interval length: 0.000028
Iteration #29	interval: [-15.000000; -14.999983]	interval length: 0.000017
Iteration #30	interval: [-15.000000; -14.999989]	interval length: 0.000011
Iteration #31	interval: [-15.000000; -14.999993]	interval length: 0.000007

Result: -15

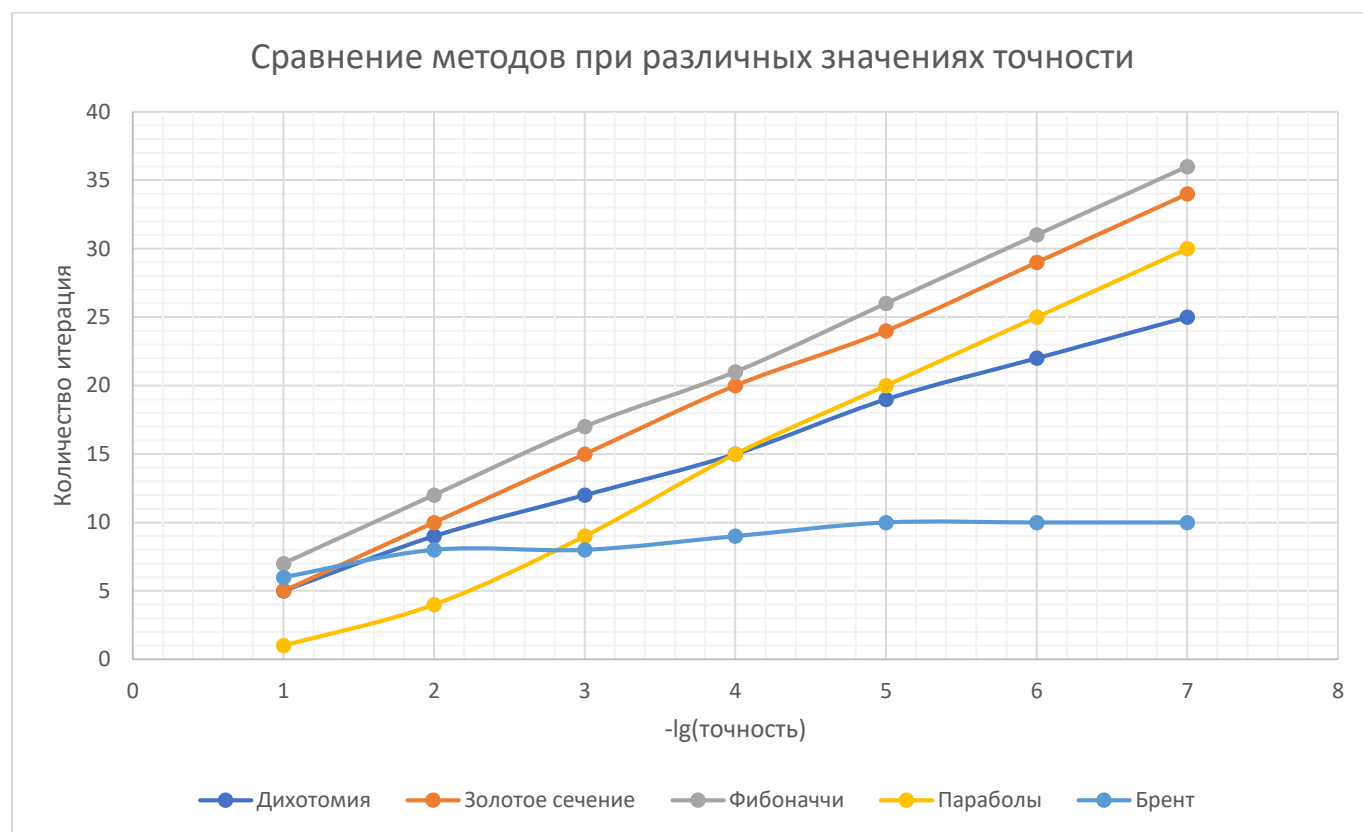
Brent:

Iteration #1	interval: [-15.000000; -5.000000]	interval length: 10.000000
Iteration #2	interval: [-15.000000; -7.360680]	interval length: 7.639320
Iteration #3	interval: [-15.000000; -11.180340]	interval length: 3.819660
Iteration #4	interval: [-15.000000; -12.082039]	interval length: 2.917961
Iteration #5	interval: [-15.000000; -13.541020]	interval length: 1.458980
Iteration #6	interval: [-15.000000; -13.885438]	interval length: 1.114562
Iteration #7	interval: [-15.000000; -14.442719]	interval length: 0.557281
Iteration #8	interval: [-15.000000; -14.574275]	interval length: 0.425725
Iteration #9	interval: [-15.000000; -14.787138]	interval length: 0.212862
Iteration #10	interval: [-15.000000; -14.837388]	interval length: 0.162612
Iteration #11	interval: [-15.000000; -14.918694]	interval length: 0.081306
Iteration #12	interval: [-15.000000; -14.937888]	interval length: 0.062112
Iteration #13	interval: [-15.000000; -14.968944]	interval length: 0.031056
Iteration #14	interval: [-15.000000; -14.976275]	interval length: 0.023725
Iteration #15	interval: [-15.000000; -14.988138]	interval length: 0.011862
Iteration #16	interval: [-15.000000; -14.990938]	interval length: 0.009062

Одномерная оптимизация нулевого порядка

Iteration #17	interval: [-15.000000; -14.995469]	interval length: 0.004531
Iteration #18	interval: [-15.000000; -14.996539]	interval length: 0.003461
Iteration #19	interval: [-15.000000; -14.998269]	interval length: 0.001731
Iteration #20	interval: [-15.000000; -14.998678]	interval length: 0.001322
Iteration #21	interval: [-15.000000; -14.999339]	interval length: 0.000661
Iteration #22	interval: [-15.000000; -14.999495]	interval length: 0.000505
Iteration #23	interval: [-15.000000; -14.999747]	interval length: 0.000253
Iteration #24	interval: [-15.000000; -14.999807]	interval length: 0.000193
Iteration #25	interval: [-15.000000; -14.999844]	interval length: 0.000156
Iteration #26	interval: [-15.000000; -14.999904]	interval length: 0.000096
Iteration #27	interval: [-15.000000; -14.999926]	interval length: 0.000074
Iteration #28	interval: [-15.000000; -14.999963]	interval length: 0.000037
Iteration #29	interval: [-15.000000; -14.999972]	interval length: 0.000028
Iteration #30	interval: [-15.000000; -14.999986]	interval length: 0.000014
Iteration #31	interval: [-15.000000; -14.999989]	interval length: 0.000011
Iteration #32	interval: [-15.000000; -14.999995]	interval length: 0.000005

Result: -15



ВЫВОД

В результате выполнения работы были исследованы и применены различные алгоритмы одномерной оптимизации нулевого порядка для исследования унимодальных и мультимодальных функций, а также сравнено количество итераций в этих алгоритмах при различных значениях точности.