Лабораторная 1

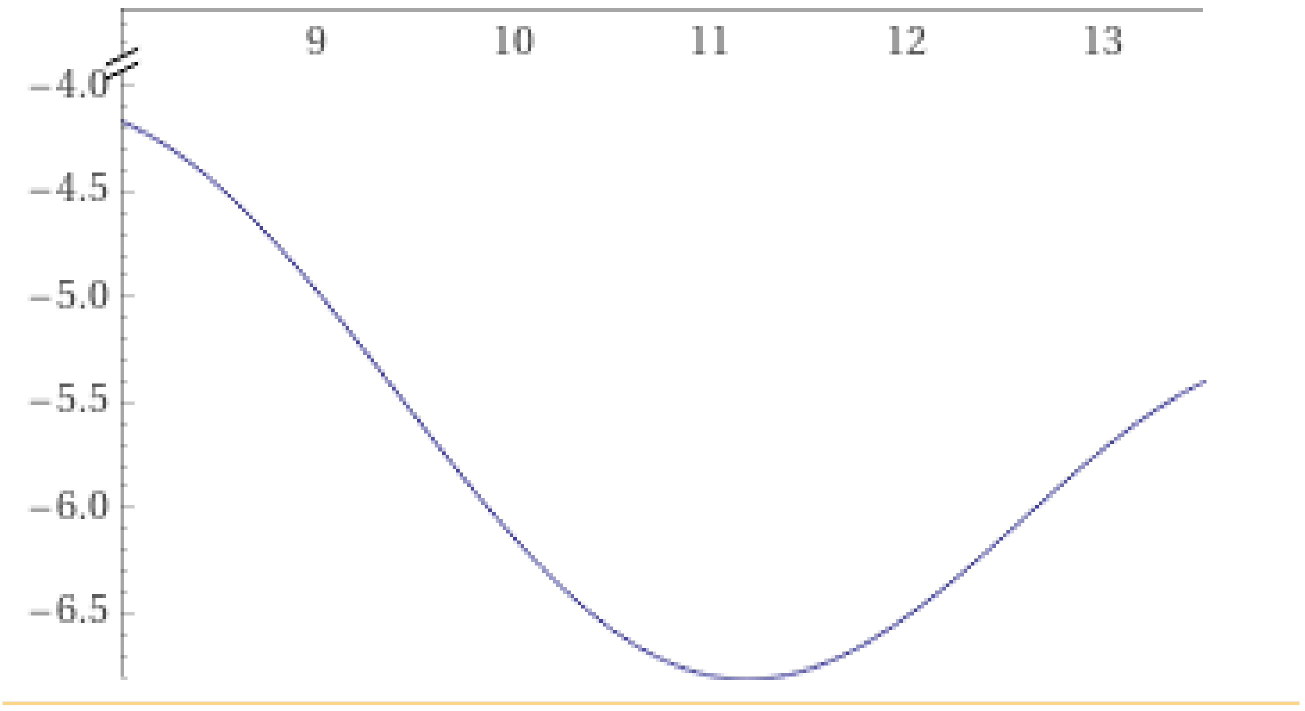
Моисеев M32001, Муров M32011

Вариант 3

Постановка задачи

Исследовать функцию $y(x) = \sin(x) - \ln x^2 - 1$ на промежутке, где она унимодальна. Найти точку минимума различными методами одномерного поиска нулевого порядка, сравнить результаты.

В качестве промежутка берем $[8, 13.5]$, а точность 1e-5



Вычислительная схема

Все методы постепенно суживают отрезок $[a_0, b_0]$, используя заданное число вычислений значений функции.

Последний отрезок и является ответом.

Также задана точность ε

Метод дихотомии

```
Выбираем $\delta < \varepsilon/2$
while (сужаем)
    $x_{1,2} = (a+b)/2 \mp \delta$
    if ($f(x_1) < f(x_2)$)
        $b = x_2$
    else
        $a = x_1$
```

Метод золотого сечения

```
$K = (3 - \sqrt{5})/2$
$w = K(b - a)$
$x_1 = a + w, x_2 = b - w$
while (сужаем)
    if ($f(x_1) < f(x_2)$)
        $b = x_2$
        $x_2 = x_1$
        $x_1 = a + b - x_1$
    else
        $a = x_1$
        $x_1 = x_2$
        $x_2 = a + b - x_2$
```

Метод Фибоначчи

$F(k)$ - k-е число фибоначчи
n - число разрешенных вычислений фнукции.
В качесте K берем $1 - \frac{F(n)}{F(n+1)}$
Далее алгоритм аналогичен методу золотого сечения

Метод парабол

```
$x_2 = (a + b)/2$
while (сужаем)
    $u = x_2 - \frac{(x_2-a)^2(f(x_2)-f(b))-(x_2-b)^2(f(x_2)-f(a))}{2[(x_2-a)(f(x_2)-f(b))-(x_2-b)(f(x_2)-f(a))]}$
    if ($f(x_2) < f(u)$)
        if ($x_2 < u$)
            $b = u$
        else
            $a = u$
    else
        $x_2 = u$
        if ($x_2 < u$)
            $a = x_2$
        else
            $b = x_2$
```

Метод Брента

```
$K = (3 - \sqrt{5})/2$
$x = w = v = (a + b)/2$
$d = e = (b - a)/2$
while (сужаем)
    $g = e$
    $e = d$
    if ($x, w, v$ разные и $f(x), f(w), f(v)$ разные)
        вычисляем $u$ как в методе парабол из $x, w, v$
    if ($a + \varepsilon \leq u \leq b - \varepsilon$ и $|u - x| < g/2$
        принимаем $u$
        $d = |u - x|$
    else if ($x < (b - a)/2$)
        $d = b - x$
        $u = x + Kd$
    else
        $d = x - a$
        $u = x - Kd$
    if ($|u - x| < \varepsilon$)
        $u = x + sign(u - x)\varepsilon$
    if ($f(u) < f(x)$)
        if ($u \geq x$)
            $a = x$
        else
```

```
            $b = x$
        $v = w$
        $w = x$
        $x = u$
    else
        if ($u \geq x$)
            $b = u$
        else
            $a = u$
        if ($f(u) \leq f(w)$ или $w = x$)
            $v = w$
            $w = u$
        else if ($f(u) \leq f(v)$)
            $v = u$
```

Результаты

		x	dx	calls	steps
4	DihotMinimizer	11.43749925	0.68750225	4	2
	GoldenMinimizer	11.55243919	0.64918694	4	3
	FibMinimizer	11.43750000	0.68750000	4	3
	ParabolaMinimizer	12.12500000	1.37500000	4	1
	BrentMinimizer	12.12500000	1.37500000	4	3
5	DihotMinimizer	11.09374963	0.34375262	6	3
	GoldenMinimizer	11.30447184	0.40121959	5	4
	FibMinimizer	11.38461538	0.42307692	5	4
	ParabolaMinimizer	11.03483804	0.28483804	5	2
	BrentMinimizer	11.06323773	0.31323773	5	4
6	DihotMinimizer	11.09374963	0.34375262	6	3
	GoldenMinimizer	11.15121959	0.24796735	6	5
	FibMinimizer	11.14285714	0.26190476	6	5
	ParabolaMinimizer	10.96896043	0.21896043	6	3
	BrentMinimizer	10.96490165	0.21490165	6	5
7	DihotMinimizer	11.26562444	0.17187781	8	4
	GoldenMinimizer	11.24593469	0.15325225	7	6
	FibMinimizer	11.23529412	0.16176471	7	6
	ParabolaMinimizer	10.96335579	0.21335579	7	4
	BrentMinimizer	10.96365829	0.21365829	7	6
8	DihotMinimizer	11.26562444	0.17187781	8	4
	GoldenMinimizer	11.18739754	0.09471510	8	7
	FibMinimizer	11.20000000	0.10000000	8	7
	ParabolaMinimizer	10.96279106	0.21279106	8	5
	BrentMinimizer	11.17638724	0.00092935	8	7
9	DihotMinimizer	11.17968703	0.08594041	10	5
	GoldenMinimizer	11.15121959	0.05853715	9	8
	FibMinimizer	11.15168539	0.06179775	9	8
	ParabolaMinimizer	10.96275665	0.21275665	9	6
	BrentMinimizer	11.17548705	0.00002915	9	8
10	DihotMinimizer	11.17968703	0.08594041	10	5
	GoldenMinimizer	11.17357879	0.03617795	10	9
	FibMinimizer	11.17013889	0.03819444	10	9

Continued on next page

Continued from previous page

		x	dx	calls	steps
	ParabolaMinimizer	10.96275343	0.21275343	10	7
	BrentMinimizer	11.17550197	0.00001423	10	9
11	DihotMinimizer	11.13671833	0.04297170	12	6
	GoldenMinimizer	11.18739754	0.02235920	11	10
	FibMinimizer	11.18669528	0.02360515	11	10
	ParabolaMinimizer	10.96275323	0.21275323	11	8
	BrentMinimizer	11.17550620	0.00001000	11	10
12	DihotMinimizer	11.13671833	0.04297170	12	6
	GoldenMinimizer	11.17885709	0.01381875	12	11
	FibMinimizer	11.18037135	0.01458886	12	11
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
13	DihotMinimizer	11.15820268	0.02148735	14	7
	GoldenMinimizer	11.17357879	0.00854046	13	12
	FibMinimizer	11.17377049	0.00901639	13	12
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
14	DihotMinimizer	11.15820268	0.02148735	14	7
	GoldenMinimizer	11.17684096	0.00527829	14	13
	FibMinimizer	11.17629179	0.00557244	14	13
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
15	DihotMinimizer	11.16894486	0.01074518	16	8
	GoldenMinimizer	11.17482483	0.00326216	15	14
	FibMinimizer	11.17532874	0.00344396	15	14
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
16	DihotMinimizer	11.16894486	0.01074518	16	8
	GoldenMinimizer	11.17607087	0.00201613	16	15
	FibMinimizer	11.17569659	0.00212848	16	15
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
17	DihotMinimizer	11.17431594	0.00537409	18	9
	GoldenMinimizer	11.17530077	0.00124604	17	16
	FibMinimizer	11.17555609	0.00131547	17	16
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
18	DihotMinimizer	11.17431594	0.00537409	18	9
	GoldenMinimizer	11.17577672	0.00077009	18	17
	FibMinimizer	11.17560976	0.00081301	18	17
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
19	DihotMinimizer	11.17700149	0.00268854	20	10
	GoldenMinimizer	11.17548257	0.00047594	19	18
	FibMinimizer	11.17558926	0.00050247	19	18
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
20	DihotMinimizer	11.17700149	0.00268854	20	10
	GoldenMinimizer	11.17566436	0.00029415	20	19
	FibMinimizer	11.17559709	0.00031054	20	19

Continued on next page

Continued from previous page

		x	dx	calls	steps
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
21	DihotMinimizer	11.17565872	0.00134577	22	11
	GoldenMinimizer	11.17555201	0.00018179	21	20
	FibMinimizer	11.17559410	0.00019193	21	20
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
22	DihotMinimizer	11.17565872	0.00134577	22	11
	GoldenMinimizer	11.17548257	0.00011235	22	21
	FibMinimizer	11.17547662	0.00011862	22	21
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
23	DihotMinimizer	11.17498733	0.00067439	24	12
	GoldenMinimizer	11.17552548	0.00006944	23	22
	FibMinimizer	11.17552149	0.00007331	23	22
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
24	DihotMinimizer	11.17498733	0.00067439	24	12
	GoldenMinimizer	11.17549896	0.00004292	24	23
	FibMinimizer	11.17550435	0.00004531	24	23
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
25	DihotMinimizer	11.17532302	0.00033869	26	13
	GoldenMinimizer	11.17551535	0.00002652	25	24
	FibMinimizer	11.17551090	0.00002800	25	24
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
26	DihotMinimizer	11.17532302	0.00033869	26	13
	GoldenMinimizer	11.17550522	0.00001639	26	25
	FibMinimizer	11.17550840	0.00001731	26	25
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
27	DihotMinimizer	11.17549087	0.00017085	28	14
	GoldenMinimizer	11.17551148	0.00001013	27	26
	FibMinimizer	11.17550935	0.00001070	27	26
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
28	DihotMinimizer	11.17549087	0.00017085	28	14
	GoldenMinimizer	11.17550761	0.00000626	28	27
	FibMinimizer	11.17551230	0.00000992	27	26
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
29	DihotMinimizer	11.17557479	0.00008692	30	15
	GoldenMinimizer	11.17550761	0.00000626	28	27
	FibMinimizer	11.17550709	0.00000613	28	27
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
30	DihotMinimizer	11.17557479	0.00008692	30	15
	GoldenMinimizer	11.17550761	0.00000626	28	27
	FibMinimizer	11.17550781	0.00000631	28	27

Continued on next page

Continued from previous page

		x	dx	calls	steps
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
31	DihotMinimizer	11.17553283	0.00004496	32	16
	GoldenMinimizer	11.17550761	0.00000626	28	27
	FibMinimizer	11.17550754	0.00000624	28	27
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
32	DihotMinimizer	11.17553283	0.00004496	32	16
	GoldenMinimizer	11.17550761	0.00000626	28	27
	FibMinimizer	11.17550764	0.00000627	28	27
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
33	DihotMinimizer	11.17551185	0.00002398	34	17
	GoldenMinimizer	11.17550761	0.00000626	28	27
	FibMinimizer	11.17550760	0.00000626	28	27
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
34	DihotMinimizer	11.17551185	0.00002398	34	17
	GoldenMinimizer	11.17550761	0.00000626	28	27
	FibMinimizer	11.17550762	0.00000626	28	27
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
35	DihotMinimizer	11.17550136	0.00001349	36	18
	GoldenMinimizer	11.17550761	0.00000626	28	27
	FibMinimizer	11.17550761	0.00000626	28	27
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
36	DihotMinimizer	11.17550136	0.00001349	36	18
	GoldenMinimizer	11.17550761	0.00000626	28	27
	FibMinimizer	11.17550761	0.00000626	28	27
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
37	DihotMinimizer	11.17550660	0.00000825	38	19
	GoldenMinimizer	11.17550761	0.00000626	28	27
	FibMinimizer	11.17550761	0.00000626	28	27
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
38	DihotMinimizer	11.17550660	0.00000825	38	19
	GoldenMinimizer	11.17550761	0.00000626	28	27
	FibMinimizer	11.17550761	0.00000626	28	27
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
39	DihotMinimizer	11.17550660	0.00000825	38	19
	GoldenMinimizer	11.17550761	0.00000626	28	27
	FibMinimizer	11.17550761	0.00000626	28	27
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
40	DihotMinimizer	11.17550660	0.00000825	38	19
	GoldenMinimizer	11.17550761	0.00000626	28	27
	FibMinimizer	11.17550761	0.00000626	28	27

Continued on next page

Continued from previous page

		x	dx	calls	steps
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
41	DihotMinimizer	11.17550660	0.00000825	38	19
	GoldenMinimizer	11.17550761	0.00000626	28	27
	FibMinimizer	11.17550761	0.00000626	28	27
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
42	DihotMinimizer	11.17550660	0.00000825	38	19
	GoldenMinimizer	11.17550761	0.00000626	28	27
	FibMinimizer	11.17550761	0.00000626	28	27
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
43	DihotMinimizer	11.17550660	0.00000825	38	19
	GoldenMinimizer	11.17550761	0.00000626	28	27
	FibMinimizer	11.17550761	0.00000626	28	27
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
44	DihotMinimizer	11.17550660	0.00000825	38	19
	GoldenMinimizer	11.17550761	0.00000626	28	27
	FibMinimizer	11.17550761	0.00000626	28	27
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
45	DihotMinimizer	11.17550660	0.00000825	38	19
	GoldenMinimizer	11.17550761	0.00000626	28	27
	FibMinimizer	11.17550761	0.00000626	28	27
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
46	DihotMinimizer	11.17550660	0.00000825	38	19
	GoldenMinimizer	11.17550761	0.00000626	28	27
	FibMinimizer	11.17550761	0.00000626	28	27
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
47	DihotMinimizer	11.17550660	0.00000825	38	19
	GoldenMinimizer	11.17550761	0.00000626	28	27
	FibMinimizer	11.17550761	0.00000626	28	27
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
48	DihotMinimizer	11.17550660	0.00000825	38	19
	GoldenMinimizer	11.17550761	0.00000626	28	27
	FibMinimizer	11.17550761	0.00000626	28	27
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10
49	DihotMinimizer	11.17550660	0.00000825	38	19
	GoldenMinimizer	11.17550761	0.00000626	28	27
	FibMinimizer	11.17550761	0.00000626	28	27
	ParabolaMinimizer	11.17550645	0.00000002	12	9
	BrentMinimizer	11.17550620	0.00001000	11	10

Метод Дихотомии достигает необходимой точности за 38 вызовов функции (19 шагов)

Методы золотого сечения и фибоначчи за 28 вызовов (27 шагов)

Метод парабол за 12 вызовов (9 шагов)

Метод Брента за 11 вызовов (10 шагов)

Выводы

Поскольку отрезок небольшой, раница между методами золотого сечения и фибоначчи незначительна. Также хорошо работает метод парабол, поскольку функция похожа на параболу, и метод Брента почти сводится к нему.

Код

Алгоритмы

```
from itertools import count
from functools import cache
from math import *
from collections import namedtuple


class CountingFunc:
    def __init__(self, f):
        self.f = f
        self.n = 0

    def __call__(self, *args, **kwargs):
        self.n += 1
        return self.f(*args, **kwargs)


MinResult = namedtuple("MinResult", ["x", "dx", "calls", "steps"])


class Minimizer:
    def __init__(self, f, a, b, eps, max_calls=0):
        self.cf = CountingFunc(f)
        self.f = cache(self.cf)
        self.a = a
        self.b = b
        self.eps = eps
        self.max_calls = max_calls or inf

    def do_step(self):
        raise NotImplementedError

    def minimize(self):
        steps = 0
        while self.cf.n < self.max_calls and self.b - self.a >= self.eps * 2:
            steps += 1
            self.do_step()
        return MinResult(x=(self.a + self.b) / 2, dx=(self.b - self.a) / 2, calls=self.cf.n, steps=step
```

```python
class DihotMinimizer(Minimizer):
    def __init__(self, *args, delta_mod=0.3, **kwargs):
        super().__init__(*args, **kwargs)
        self.delta = self.eps * delta_mod

    @staticmethod
    def next_step(f, delta, a, b):
        mid = (a + b) / 2
        x1, x2 = mid - delta, mid + delta
        if f(x1) < f(x2):
            b = x2
        else:
            a = x1
        return a, b

    def do_step(self):
        self.a, self.b = self.next_step(self.f, self.delta, self.a, self.b)


class GoldenMinimizer(Minimizer):
    q = (3 - sqrt(5)) / 2

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        w = self.q * (self.b - self.a)
        self.x1 = self.a + w
        self.x2 = self.b - w

    @staticmethod
    def next_step(f, a, b, x1, x2):
        if f(x1) < f(x2):
            b, x2 = x2, x1
            x1 = a + b - x2
        else:
            a, x1 = x1, x2
            x2 = a + b - x1
        return a, b, x1, x2

    def do_step(self):
        self.a, self.b, self.x1, self.x2 = self.next_step(self.f, self.a, self.b, self.x1, self.x2)


@cache
def fib(n):
    if n < 2:
        return 1
    return fib(n - 1) + fib(n - 2)


class FibMinimizer(GoldenMinimizer):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        if isinf(self.max_calls):
            n = next(k for k in count() if self.b - self.a < self.eps * fib(k + 2))
        else:
            n = self.max_calls
        w = (self.b - self.a) * fib(n) / fib(n + 1)
        self.x1 = self.b - w
        self.x2 = self.a + w


class ParabolaMinimizer(Minimizer):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.x2 = (self.a + self.b) / 2

    @staticmethod
    def next_step(f, a, x2, b):
        u = x2 - ((x2 - a) ** 2 * (f(x2) - f(b)) - (x2 - b) ** 2 * (f(x2) - f(a))) / (
            2 * ((x2 - a) * (f(x2) - f(b)) - (x2 - b) * (f(x2) - f(a)))
        )
        if f(x2) < f(u):
            if x2 < u:
                b = u
            else:
                a = u
        else:
            if x2 < u:
                a, x2 = x2, u
            else:
                x2, b = u, x2
        return a, x2, b

    def do_step(self):
        self.a, self.x2, self.b = self.next_step(self.f, self.a, self.x2, self.b)


class BrentMinimizer(Minimizer):
    k = (3 - sqrt(5)) / 2

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.x = self.w = self.v = (self.a + self.b) / 2
        self.d = self.e = self.b - self.a

    @staticmethod
    def next_step(f, x, w, v, d, e, a, b, k, eps):
        g, e = e, d
        u = None
        if len({x, w, v}) == 3 and len(set(map(f, [x, w, v]))):
            x1, x2, x3 = sorted([x, w, v])
            u = x2 - ((x2 - x1) ** 2 * (f(x2) - f(x3)) - (x2 - x3) ** 2 * (f(x2) - f(x1))) / (
                2 * ((x2 - x1) * (f(x2) - f(x3)) - (x2 - x3) * (f(x2) - f(x1)))
            )
            if a + eps <= u <= b - eps and abs(u - x) < g / 2:
                d = abs(u - x)
            else:
                u = None
        if u is None:
            if x < (b - a) / 2:
                u = x + k * (b - x)
                d = b - x
            else:
                u = x - k * (x - a)
                d = x - a
        if abs(u - x) < eps:
            u = x + copysign(eps, u - x)
        if f(u) <= f(x):
            if u >= x:
                a = x
            else:
                b = x
            v, w, x = w, x, u
        else:
            if u >= x:
                b = u
            else:
                a = u
            if f(u) <= f(w) or w == x:
                v, w = w, u
            elif f(u) <= f(v):
                v = u
        return a, b, x, w, v, d, e

    def do_step(self):
        self.a, self.b, self.x, self.w, self.v, self.d, self.e = self.next_step(
            self.f, self.x, self.w, self.v, self.d, self.e, self.a, self.b, self.k, self.eps
        )
```

Работа

```
from primat import *
from org_table import table
import pandas as pd

f = lambda x: sin(x) - log(x**2) - 1
minimizers = [DihotMinimizer, GoldenMinimizer, FibMinimizer, ParabolaMinimizer, BrentMinimizer]


def test(n, minimizer):
    return minimizer(f, 8, 13.5, 10 ** (-5), max_calls=n).minimize()


ns = range(4, 50)
index = pd.MultiIndex.from_product([ns, [m.__name__ for m in minimizers]], names=["n", "Minimizer"])
df = pd.DataFrame([test(n, m) for n in ns for m in minimizers], index=index)
with open("res.org", "w") as file:
    print(table(df, fmt=".8f"), file=file)
```