



Министерство науки и высшего образования Российской Федерации  
Калужский филиал федерального государственного автономного  
образовательного учреждения высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(КФ МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ ИУК Информатика и управление

КАФЕДРА ИУК4 Программное обеспечение ЭВМ, информационные технологии

## ЛАБОРАТОРНАЯ РАБОТА

### «Минимизация функций»

по дисциплине: *«Методы принятия решений в программной инженерии»*

Выполнил: студент группы ИУК4-72Б

\_\_\_\_\_  
(Подпись)

Губин Е.В.

\_\_\_\_\_  
(И.О. Фамилия)

Проверил:

\_\_\_\_\_  
(Подпись)

Никитенко У.В.

\_\_\_\_\_  
(И.О. Фамилия)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2025

**Цель:** ознакомиться с методами одномерного поиска, используемыми в методах минимизации функций. Сравнить различные алгоритмы по эффективности на тестовых примерах.

**Задание (вариант №23):**

Найти минимум и максимум унимодальной на отрезке  $[a, b]$  функции  $f(x)$  с точностью  $E$ .

23

$$x^3 - e^x$$

-1

0

**Результаты выполнения работы:**

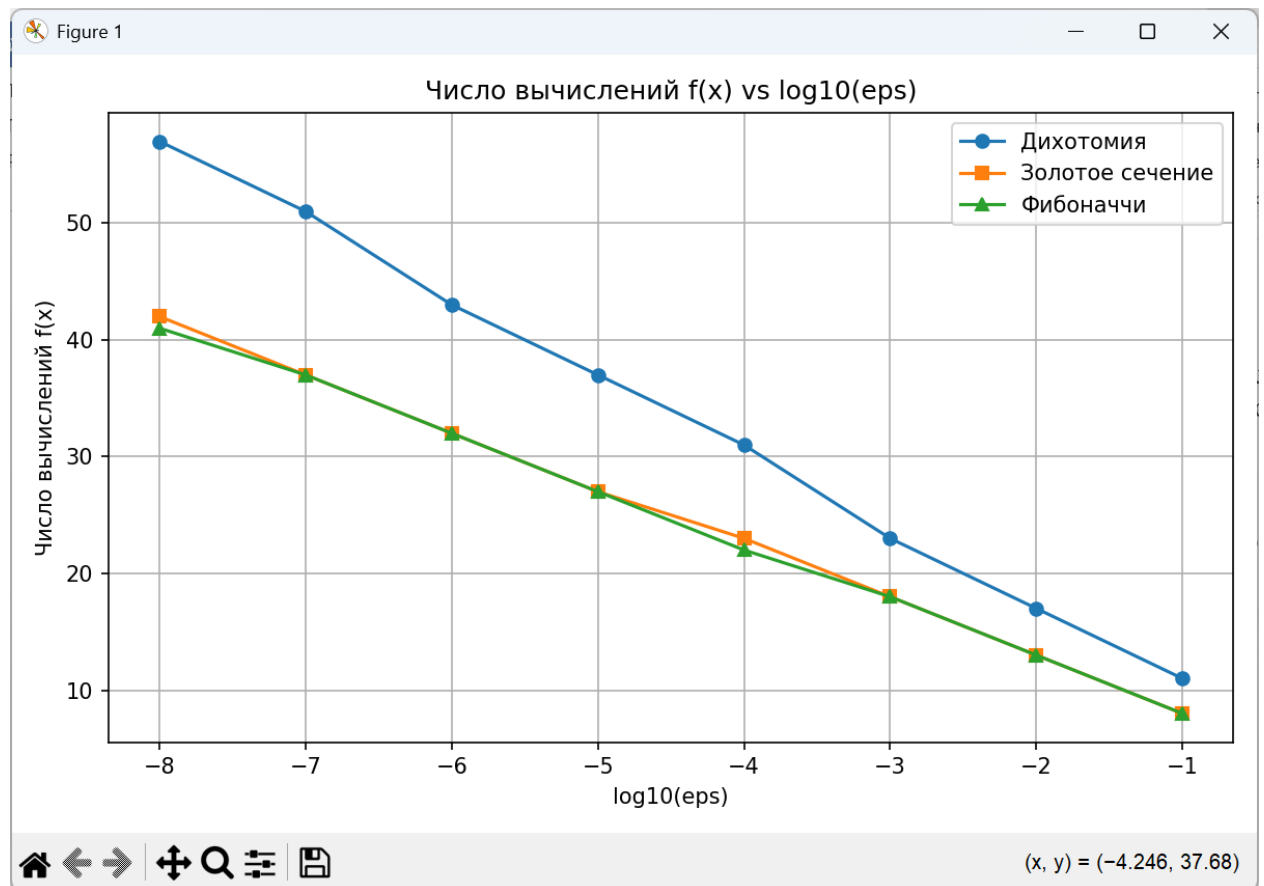


Рисунок 1 График зависимости количества вычислений целевой функции от логарифма задаваемой точности  $E$

```

Дихотомия: x_min = -0.999999511582, f = -1.367878155596e+00, f-vals = 43
Таблица итераций сохранена: dichotomy_history.csv (строк 21)
Золотое сечение: x_min = -0.999999565161, f = -1.367878296624e+00, f-vals = 32
Таблица итераций сохранена: golden_history.csv (строк 29)
Фибоначчи: x_min = -0.999998885810, f = -1.367876508492e+00, f-vals = 32
Таблица итераций сохранена: fibonacci_history.csv (строк 29)
    
```

Рисунок 2 Результаты выполнения программы (поиск минимума функции различными методами)

Таблица 1 Результаты исследования метода дихотомии

| k  | a    | b         | length     | ratio_prev_to_curr | x_test1             | f_x_test1         | x_test2       | f_x_test2      | cumulative_f_calls |
|----|------|-----------|------------|--------------------|---------------------|-------------------|---------------|----------------|--------------------|
| 1  | -1.0 | -0.499999 | 0.50000025 |                    | -0.50000025         | -0.73153069558008 | -0.49999975   | -0.73153062384 | 2                  |
| 2  | -1.0 | -0.749999 | 0.2500003  | 1.999998000002999  | -0.750000125        | -0.89424170463273 | -0.749999625  | -0.89424109706 | 4                  |
| 3  | -1.0 | -0.874999 | 0.1250004  | 1.999996000014000  | -0.8750000625000001 | -1.08678401217933 | -0.8749995625 | -1.08678307217 | 6                  |
| 4  | -1.0 | -0.937499 | 0.0625004  | 1.999992000059998  | -0.93750003125      | -1.21558030621158 | -0.9374995312 | -1.21557918365 | 8                  |
| 5  | -1.0 | -0.968749 | 0.0312504  | 1.999984000247990  | -0.968750015625     | -1.28870639616547 | -0.9687495156 | -1.28870517822 | 10                 |
| 6  | -1.0 | -0.984374 | 0.0156254  | 1.999968001007963  | -0.9843750078125    | -1.32752632637525 | -0.9843745078 | -1.32752505972 | 12                 |
| 7  | -1.0 | -0.992187 | 0.0078129  | 1.999936004063733  | -0.99218750390625   | -1.34750989410845 | -0.9921870039 | -1.34750860285 | 14                 |
| 8  | -1.0 | -0.996093 | 0.0039067  | 1.999872016317901  | -0.996093751953125  | -1.35764625244894 | -0.9960932519 | -1.35764494886 | 16                 |
| 9  | -1.0 | -0.998046 | 0.0019536  | 1.999744065391256  | -0.9980468759765625 | -1.36275072203588 | -0.9980463759 | -1.36274941218 | 18                 |
| 10 | -1.0 | -0.999022 | 0.0009770  | 1.999488261753927  | -0.9990234379882812 | -1.36531204778772 | -0.9990229379 | -1.36531073485 | 20                 |
| 11 | -1.0 | -0.999511 | 0.0004887  | 1.998977046992260  | -0.9995117189941406 | -1.36659498569778 | -0.9995112189 | -1.36659367115 | 22                 |
| 12 | -1.0 | -0.999755 | 0.0002446  | 1.997956184711971  | -0.9997558594970704 | -1.36723702369818 | -0.9997553594 | -1.36723570841 | 24                 |
| 13 | -1.0 | -0.999877 | 0.0001225  | 1.995920706744955  | -0.9998779297485352 | -1.36755818499558 | -0.9998774297 | -1.36755686932 | 26                 |
| 14 | -1.0 | -0.999938 | 6.1535125  | 1.991874559544355  | -0.9999389648742676 | -1.36771880122306 | -0.9999384648 | -1.36771748535 | 28                 |
| 15 | -1.0 | -0.999968 | 3.1017562  | 1.983880100373278  | -0.9999694824371338 | -1.36779911823206 | -0.9999689824 | -1.36779780226 | 30                 |
| 16 | -1.0 | -0.999984 | 1.5758781  | 1.968271658434952  | -0.999984741218567  | -1.36783927896044 | -0.9999842412 | -1.36783796294 | 32                 |
| 17 | -1.0 | -0.999991 | 8.1293907  | 1.938494775623307  | -0.9999923706092835 | -1.36785935988061 | -0.9999918706 | -1.36785804384 | 34                 |
| 18 | -1.0 | -0.999995 | 4.3146953  | 1.884116963413330  | -0.9999961853046417 | -1.36786940047969 | -0.9999956853 | -1.36786808443 | 36                 |
| 19 | -1.0 | -0.999997 | 2.4073476  | 1.792302539267664  | -0.9999980926523209 | -1.36787442081398 | -0.9999975926 | -1.36787310476 | 38                 |
| 20 | -1.0 | -0.999998 | 1.4536738  | 1.656043889325742  | -0.9999990463261604 | -1.36787693098981 | -0.9999985463 | -1.36787561493 | 40                 |
| 21 | -1.0 | -0.999999 | 9.7683691  | 1.488143834630331  | -0.9999995231630803 | -1.36787818607990 | -0.9999990231 | -1.36787687002 | 42                 |

Таблица 2 Результаты исследования метода золотого сечения

| k  | a    | b         | length            | ratio_prev | x_test1   | f_x_test1 | x_test2   | f_x_test2 | cumulative_f_calls |
|----|------|-----------|-------------------|------------|-----------|-----------|-----------|-----------|--------------------|
| 1  | -1.0 | -0.381966 | 0.618033988749894 |            | -0.763932 | -0.911655 | -0.618033 | -0.775071 | 3                  |
| 2  | -1.0 | -0.618033 | 0.381966          | 1.6180339  | -0.854101 | -1.048724 | -0.763932 | -0.911655 | 4                  |
| 3  | -1.0 | -0.763932 | 0.236067          | 1.6180339  | -0.909830 | -1.155741 | -0.854101 | -1.048724 | 5                  |
| 4  | -1.0 | -0.854101 | 0.145898          | 1.6180339  | -0.944271 | -1.230922 | -0.909830 | -1.155741 | 6                  |
| 5  | -1.0 | -0.909830 | 0.090169          | 1.6180339  | -0.965558 | -1.280962 | -0.944271 | -1.230922 | 7                  |
| 6  | -1.0 | -0.944271 | 0.055728          | 1.6180339  | -0.978713 | -1.313285 | -0.965558 | -1.280962 | 8                  |
| 7  | -1.0 | -0.965558 | 0.034441          | 1.6180339  | -0.986844 | -1.333801 | -0.978713 | -1.313285 | 9                  |
| 8  | -1.0 | -0.978713 | 0.021286          | 1.6180339  | -0.991869 | -1.346688 | -0.986844 | -1.333801 | 10                 |
| 9  | -1.0 | -0.986844 | 0.013155          | 1.6180339  | -0.994975 | -1.354733 | -0.991869 | -1.346688 | 11                 |
| 10 | -1.0 | -0.991869 | 0.008130          | 1.6180339  | -0.996894 | -1.359735 | -0.994975 | -1.354733 | 12                 |
| 11 | -1.0 | -0.994975 | 0.005024          | 1.6180339  | -0.998080 | -1.362839 | -0.996894 | -1.359735 | 13                 |
| 12 | -1.0 | -0.996894 | 0.003105          | 1.6180339  | -0.998813 | -1.364761 | -0.998080 | -1.362839 | 14                 |
| 13 | -1.0 | -0.998080 | 0.001919          | 1.6180339  | -0.999266 | -1.365951 | -0.998813 | -1.364761 | 15                 |
| 14 | -1.0 | -0.998813 | 0.001186          | 1.6180339  | -0.999546 | -1.366687 | -0.999266 | -1.365951 | 16                 |
| 15 | -1.0 | -0.999266 | 0.000733          | 1.6180339  | -0.999719 | -1.367142 | -0.999546 | -1.366687 | 17                 |
| 16 | -1.0 | -0.999546 | 0.000453          | 1.6180339  | -0.999826 | -1.367423 | -0.999719 | -1.367142 | 18                 |
| 17 | -1.0 | -0.999719 | 0.000280          | 1.6180339  | -0.999893 | -1.367597 | -0.999826 | -1.367423 | 19                 |
| 18 | -1.0 | -0.999826 | 0.000173          | 1.6180339  | -0.999933 | -1.367705 | -0.999893 | -1.367597 | 20                 |
| 19 | -1.0 | -0.999893 | 0.000106          | 1.6180339  | -0.999959 | -1.367771 | -0.999933 | -1.367705 | 21                 |
| 20 | -1.0 | -0.999933 | 6.610696          | 1.6180339  | -0.999974 | -1.367812 | -0.999959 | -1.367771 | 22                 |
| 21 | -1.0 | -0.999959 | 4.085634          | 1.6180339  | -0.999984 | -1.367838 | -0.999974 | -1.367812 | 23                 |
| 22 | -1.0 | -0.999974 | 2.525061          | 1.6180339  | -0.999990 | -1.367854 | -0.999984 | -1.367838 | 24                 |
| 23 | -1.0 | -0.999984 | 1.560573          | 1.6180339  | -0.999994 | -1.367863 | -0.999990 | -1.367854 | 25                 |
| 24 | -1.0 | -0.999990 | 9.644875          | 1.6180339  | -0.999996 | -1.367869 | -0.999994 | -1.367863 | 26                 |
| 25 | -1.0 | -0.999994 | 5.960860          | 1.6180339  | -0.999997 | -1.367873 | -0.999996 | -1.367869 | 27                 |
| 26 | -1.0 | -0.999996 | 3.684014          | 1.6180339  | -0.999998 | -1.367875 | -0.999997 | -1.367873 | 28                 |
| 27 | -1.0 | -0.999997 | 2.276846          | 1.6180339  | -0.999999 | -1.367877 | -0.999998 | -1.367875 | 29                 |
| 28 | -1.0 | -0.999998 | 1.407168          | 1.6180339  | -0.999999 | -1.367878 | -0.999999 | -1.367877 | 30                 |
| 29 | -1.0 | -0.999999 | 8.696778          | 1.6180339  | -0.999999 | -1.367878 | -0.999999 | -1.367878 | 31                 |

Таблица 3 Результаты исследования метода Фибоначчи

| k  | a         | b         | length             | ratio_prev_to_curr  | x_test1       | f_x_test1       | x_test2        | f_x_test2 | cumulative_f_calls |
|----|-----------|-----------|--------------------|---------------------|---------------|-----------------|----------------|-----------|--------------------|
| 1  | -1.0      | -0.381966 | 0.6180339887496481 |                     | -0.7639320225 | -0.911655882618 | -0.61803398874 | -0.775071 | 3                  |
| 2  | -1.0      | -0.618033 | 0.3819660          | 1.6180339887482036  | -0.8541019662 | -1.048724268769 | -0.76393202250 | -0.911655 | 4                  |
| 3  | -1.0      | -0.763932 | 0.2360679          | 1.6180339887543225  | -0.9098300562 | -1.155741523908 | -0.85410196624 | -1.048724 | 5                  |
| 4  | -1.0      | -0.854101 | 0.1458980          | 1.6180339887383028  | -0.9442719099 | -1.230922187424 | -0.90983005625 | -1.155741 | 6                  |
| 5  | -1.0      | -0.909830 | 0.0901699          | 1.6180339887802428  | -0.9655581462 | -1.280962921008 | -0.94427190999 | -1.230922 | 7                  |
| 6  | -1.0      | -0.944271 | 0.0557280          | 1.6180339886704427  | -0.9787137637 | -1.313285106099 | -0.96555814625 | -1.280962 | 8                  |
| 7  | -1.0      | -0.965558 | 0.0344418          | 1.6180339889579034  | -0.9868443825 | -1.333801178443 | -0.97871376374 | -1.313285 | 9                  |
| 8  | -1.0      | -0.978713 | 0.0212862          | 1.6180339882053216  | -0.9918693812 | -1.346688648465 | -0.98684438251 | -1.333801 | 10                 |
| 9  | -1.0      | -0.986844 | 0.0131556          | 1.6180339901756062  | -0.9949750012 | -1.354733316067 | -0.99186938123 | -1.346688 | 11                 |
| 10 | -1.0      | -0.991869 | 0.0081306          | 1.6180339850173562  | -0.9968943799 | -1.359735755376 | -0.99497500128 | -1.354733 | 12                 |
| 11 | -1.0      | -0.994975 | 0.0050249          | 1.6180339985218082  | -0.9980806213 | -1.362839128161 | -0.99689437994 | -1.359735 | 13                 |
| 12 | -1.0      | -0.996894 | 0.0031056          | 1.6180339631666938  | -0.9988137586 | -1.364761589613 | -0.99808062133 | -1.362839 | 14                 |
| 13 | -1.0      | -0.998080 | 0.0019193          | 1.6180340557275876  | -0.9992668627 | -1.365951446417 | -0.99881375861 | -1.364761 | 15                 |
| 14 | -1.0      | -0.998813 | 0.0011862          | 1.6180338134000378  | -0.9995468959 | -1.366687470147 | -0.99926686271 | -1.365951 | 16                 |
| 15 | -1.0      | -0.999266 | 0.0007331          | 1.6180344478216657  | -0.9997199668 | -1.367142609719 | -0.99954689590 | -1.366687 | 17                 |
| 16 | -1.0      | -0.999546 | 0.0004531          | 1.6180327868852882  | -0.9998269290 | -1.367423993038 | -0.99971996681 | -1.367142 | 18                 |
| 17 | -1.0      | -0.999719 | 0.0002800          | 1.6180371352784042  | -0.9998930377 | -1.367597939984 | -0.99982692909 | -1.367423 | 19                 |
| 18 | -1.0      | -0.999826 | 0.0001730          | 1.6180257510732505  | -0.9999338913 | -1.367705449200 | -0.99989303772 | -1.367597 | 20                 |
| 19 | -1.0      | -0.999893 | 0.0001069          | 1.6180555555547986  | -0.9999591463 | -1.367771914758 | -0.99993389136 | -1.367705 | 21                 |
| 20 | -1.0      | -0.999933 | 6.6108630          | 1.617977528091869   | -0.9999747450 | -1.367812969043 | -0.99995914635 | -1.367771 | 22                 |
| 21 | -1.0      | -0.999959 | 4.0853648          | 1.6181818181810277  | -0.9999844013 | -1.367838384377 | -0.99997474501 | -1.367812 | 23                 |
| 22 | -1.0      | -0.999974 | 2.5254982          | 1.6176470588255982  | -0.9999903436 | -1.367854024878 | -0.99998440133 | -1.367838 | 24                 |
| 23 | -1.0      | -0.999984 | 1.5598665          | 1.6190476190421963  | -0.9999940576 | -1.367863800305 | -0.99999034368 | -1.367854 | 25                 |
| 24 | -1.0      | -0.999990 | 9.6563168          | 1.615384615398766   | -0.9999962860 | -1.367869665603 | -0.99999405765 | -1.367863 | 26                 |
| 25 | -1.0      | -0.999994 | 5.9423488          | 1.6249999999929938  | -0.9999977716 | -1.367873575820 | -0.99999628603 | -1.367869 | 27                 |
| 26 | -1.0      | -0.999996 | 3.7139680          | 1.600000000017936   | -0.9999985144 | -1.367875530933 | -0.99999777161 | -1.367873 | 28                 |
| 27 | -1.0      | -0.999997 | 2.2283808          | 1.66666666666168448 | -0.9999992572 | -1.367877486050 | -0.99999851441 | -1.367875 | 29                 |
| 28 | -1.0      | -0.999998 | 1.4855872          | 01.май              | -0.9999992572 | -1.367877486050 | -0.99999925720 | -1.367877 | 30                 |
| 29 | -0.999999 | -0.999998 | 7.4279360          | 2.0                 | -0.9999992572 | -1.367877486050 | -0.99999851441 | -1.367875 | 31                 |

## Листинг программы:

```
import math
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

class FuncCounter:
    def __init__(self, f):
        self._f = f
        self.count = 0
    def __call__(self, x):
        self.count += 1
        return self._f(x)
    def reset(self):
        self.count = 0

def f_raw(x):
    return x**3 - math.exp(x)

def history_to_df(history):
    df = pd.DataFrame(history)
    lengths = df['length'].values
    ratios = [np.nan]
    for i in range(1, len(lengths)):
        prev = lengths[i-1]
        curr = lengths[i]
        ratios.append(prev / curr if curr != 0 else np.nan)
    df['ratio_prev_to_curr'] = ratios
    cols = ['k', 'a', 'b', 'length', 'ratio_prev_to_curr',
            'x_test1', 'f_x_test1', 'x_test2', 'f_x_test2',
            'cumulative_f_calls']
```

```

cols_present = [c for c in cols if c in df.columns]
return df[cols_present]

def dichotomy_with_history(a, b, eps, f, delta=None):
    if delta is None:
        delta = eps / 4.0
    history = []
    while (b - a) > eps:
        xm = (a + b) / 2.0
        x1 = xm - delta
        x2 = xm + delta
        f1 = f(x1)
        f2 = f(x2)
        if f1 < f2:
            b = x2
        else:
            a = x1
        length = b - a
        history.append({
            'k': len(history) + 1,
            'a': a, 'b': b, 'length': length,
            'x_test1': x1, 'f_x_test1': f1,
            'x_test2': x2, 'f_x_test2': f2,
            'cumulative_f_calls': f.count
        })
    x_min = (a + b) / 2.0
    return x_min, f(x_min), history

def golden_with_history(a, b, eps, f):
    phi = (math.sqrt(5) - 1) / 2.0
    x1 = b - phi * (b - a)
    x2 = a + phi * (b - a)
    f1 = f(x1)
    f2 = f(x2)
    history = []
    while (b - a) > eps:
        if f1 < f2:
            b = x2
            x2 = x1
            f2 = f1
            x1 = b - phi * (b - a)
            f1 = f(x1)
        else:
            a = x1
            x1 = x2
            f1 = f2
            x2 = a + phi * (b - a)
            f2 = f(x2)
        length = b - a
        history.append({
            'k': len(history) + 1,
            'a': a, 'b': b, 'length': length,
            'x_test1': x1, 'f_x_test1': f1,
            'x_test2': x2, 'f_x_test2': f2,
            'cumulative_f_calls': f.count
        })
    x_min = (a + b) / 2.0
    return x_min, f(x_min), history

def fibonacci_with_history(a, b, eps, f):
    L0 = b - a
    F = [1, 1]
    while F[-1] < L0 / eps:
        F.append(F[-1] + F[-2])

```

```

n = len(F)
history = []
if n < 3:
    x_mid = (a + b) / 2.0
    val = f(x_mid)
    history.append({
        'k': 1, 'a': a, 'b': b, 'length': b-a,
        'x_test1': x_mid, 'f_x_test1': val,
        'x_test2': np.nan, 'f_x_test2': np.nan,
        'cumulative_f_calls': f.count
    })
    return x_mid, val, history
x1 = a + (F[-3] / F[-1]) * (b - a)
x2 = a + (F[-2] / F[-1]) * (b - a)
f1 = f(x1)
f2 = f(x2)
k = 1
while k <= n - 2 and (b - a) > eps:
    if f1 < f2:
        b = x2
        x2 = x1
        f2 = f1
        idx = n - k - 3
        if idx >= 0:
            x1 = a + (F[idx] / F[n - k - 1]) * (b - a)
            f1 = f(x1)
        else:
            x1 = (a + b) / 2.0
            f1 = f(x1)
    else:
        a = x1
        x1 = x2
        f1 = f2
        idx = n - k - 2
        if idx >= 0:
            x2 = a + (F[idx] / F[n - k - 1]) * (b - a)
            f2 = f(x2)
        else:
            x2 = (a + b) / 2.0
            f2 = f(x2)
    length = b - a
    history.append({
        'k': k,
        'a': a, 'b': b, 'length': length,
        'x_test1': x1, 'f_x_test1': f1,
        'x_test2': x2, 'f_x_test2': f2,
        'cumulative_f_calls': f.count
    })
    k += 1
x_min = (a + b) / 2.0
return x_min, f(x_min), history

def main():
    a0 = -1.0
    b0 = 0.0
    eps0 = 1e-6
    fc = FuncCounter(f_raw)
    x_d, fx_d, hist_d = dichotomy_with_history(a0, b0, eps0, fc)
    df_d = history_to_df(hist_d)
    df_d.to_csv("dichotomy_history.csv", sep=";", index=False)
    print(f"Дихотомия: x_min = {x_d:.12f}, f = {fx_d:.12e}, f-evals = {fc.count}")
    print(f"Таблица итераций сохранена: dichotomy_history.csv (строк {len(df_d)})")

```

```

fc = FuncCounter(f_raw)
x_g, fx_g, hist_g = golden_with_history(a0, b0, eps0, fc)
df_g = history_to_df(hist_g)
df_g.to_csv("golden_history.csv", sep=";", index=False)
print(f"Золотое сечение: x_min = {x_g:.12f}, f = {fx_g:.12e}, f-evals =
{fc.count}")
print(f"Таблица итераций сохранена: golden_history.csv (строк
{len(df_g)})")
fc = FuncCounter(f_raw)
x_f, fx_f, hist_f = fibonacci_with_history(a0, b0, eps0, fc)
df_f = history_to_df(hist_f)
df_f.to_csv("fibonacci_history.csv", sep=";", index=False)
print(f"Фибоначчи: x_min = {x_f:.12f}, f = {fx_f:.12e}, f-evals =
{fc.count}")
print(f"Таблица итераций сохранена: fibonacci_history.csv (строк
{len(df_f)})")
print("\nПервые 6 итераций (Дихотомия):")
print(df_d.head(6).to_string(index=False))
print("\nПервые 6 итераций (Золотое сечение):")
print(df_g.head(6).to_string(index=False))
print("\nПервые 6 итераций (Фибоначчи):")
print(df_f.head(6).to_string(index=False))
plt.figure(figsize=(8,5))
plt.plot(df_d['k'], df_d['length'], marker='o', label='Дихотомия')
plt.plot(df_g['k'], df_g['length'], marker='s', label='Золотое сечение')
plt.plot(df_f['k'], df_f['length'], marker='^', label='Фибоначчи')
plt.yscale('log')
plt.xlabel('k (итерация)')
plt.ylabel('Длина интервала (log scale)')
plt.title('Длина интервала по итерациям (eps = {}).format(eps0))
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.savefig("interval_lengths_by_iteration.png")
print("\nГрафик длины интервала сохранён:
interval_lengths_by_iteration.png")

if __name__ == "__main__":
    main

```

**Выводы:** в ходе лабораторной работы я реализовал на языке Python три метода одномерной минимизации функции: метод дихотомии, метод золотого сечения и метод Фибоначчи. На основе построенных таблиц итераций и графиков были проанализированы границы интервалов неопределённости на каждом шаге, длины интервалов и их уменьшение, а также количество вычислений функции.