МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Московский государственный технический университет

им. Н.Э. Баумана

(МГТУ им. Н.Э. Баумана)

Кафедра «Информационная безопасность» (ИУ8)

Лабораторная работа № 1

Исследование методов прямого поиска экстремума унимодальной функции одного переменного

Выполнила: Янгирова И.А.,

студентка группы ИУ8-33

Проверил: Строганов И.С.,

преподаватель каф. ИУ8

г. Москва 2020 г.

Цель работы

Исследовать функционирование и провести сравнительный анализ различных алгоритмов прямого поиска экстремума (пассивный поиск, метод дихотомии, золотого сечения, Фибоначчи) на примере унимодальной функции одного переменного.

Постановка задачи

На интервале [a,b] задана унимодальная функция одного переменного f (x) . Используя методы последовательного поиска (дихотомии, золотого сечения и Фибоначчи), найти интервал нахождения минимума f(x) при заданной наибольшей допустимой длине интервала неопределенности ε = 0,1. Провести сравнение с методом оптимального пассивного поиска (п. 2.2 в [1]). Результат, в зависимости от числа точек разбиения N, представить в виде таблицы.

Вариант 20:

№пп	Функция $f(x)$	a	b	Метод поиска	
20	$x^2 \exp[\sin(x)]$	16	20	опт. пассивный	золотое сечение

График функции

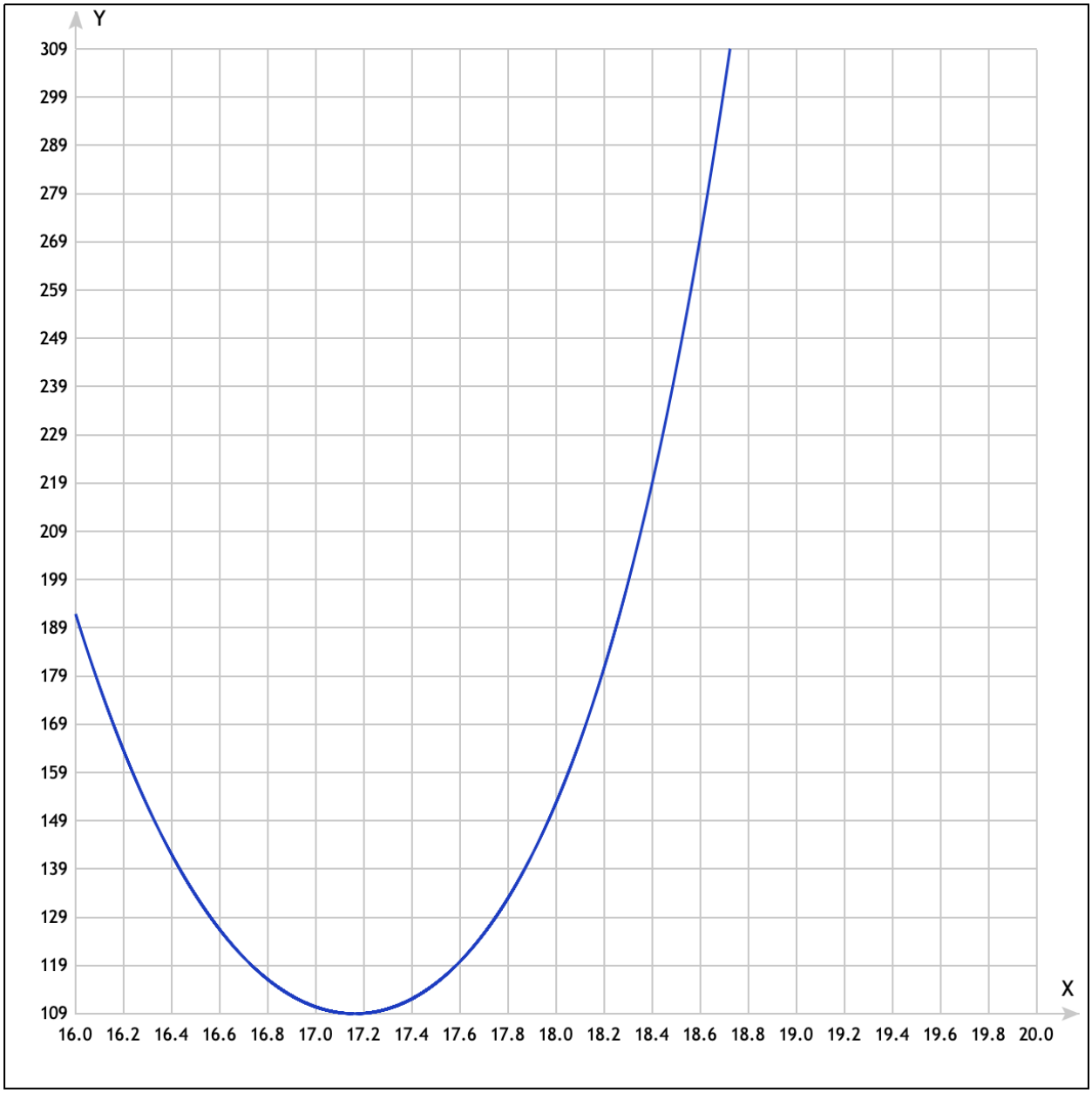

Рис.1. График функции

Скриншот консоли

Optimal Passive Search Method
accuracy = 0.05

№	Значение x	Значение f(x)
1	16.05	184.198
2	16.1	176.897
3	16.15	170.044
4	16.2	163.629
5	16.25	157.638
6	16.3	152.06
7	16.35	146.882
8	16.4	142.089
9	16.45	137.67
10	16.5	133.611
11	16.55	129.901
12	16.6	126.527
13	16.65	123.478
14	16.7	120.744
15	16.75	118.316
16	16.8	116.186
17	16.85	114.345
18	16.9	112.787
19	16.95	111.508
20	17	110.502
21	17.05	109.766
22	17.1	109.3
23	17.15	109.101
24	17.2	109.171
25	17.25	109.512
26	17.3	110.127
27	17.35	111.021
28	17.4	112.2
29	17.45	113.671
30	17.5	115.443
31	17.55	117.527
32	17.6	119.935
33	17.65	122.682
34	17.7	125.782
35	17.75	129.252
36	17.8	133.113
37	17.85	137.383
38	17.9	142.088
39	17.95	147.25
40	18	152.896
41	18.05	159.054
42	18.1	165.755
43	18.15	173.03
44	18.2	180.912
45	18.25	189.437
46	18.3	198.64
47	18.35	208.559
48	18.4	219.233
49	18.45	230.7
50	18.5	243

41	18.05	159.054
42	18.1	165.755
43	18.15	173.03
44	18.2	180.912
45	18.25	189.437
46	18.3	198.64
47	18.35	208.559
48	18.4	219.233
49	18.45	230.7
50	18.5	243
51	18.55	256.171
52	18.6	270.251
53	18.65	285.276
54	18.7	301.282
55	18.75	318.301
56	18.8	336.359
57	18.85	355.48
58	18.9	375.683
59	18.95	396.979
60	19	419.371
61	19.05	442.853
62	19.1	467.412
63	19.15	493.02
64	19.2	519.64
65	19.25	547.219
66	19.3	575.694
67	19.35	604.983
68	19.4	634.992
69	19.45	665.609
70	19.5	696.71
71	19.55	728.151
72	19.6	759.775
73	19.65	791.413
74	19.7	822.88
75	19.75	853.982
76	19.8	884.513
77	19.85	914.264
78	19.9	943.016
79	19.95	970.554
80	20	996.66

Iteration amount = 80
Received uncertainty interval length = 0.1
Function extremum: (17.15 ; 109.101)

```
    Golden Ratio Method
-------------------------------------------------------------------------------------------
|  №  | Значение x1 | Значение x2 | Значение f(x1) | Значение f(x2) | Длина интервала |
-------------------------------------------------------------------------------------------
|  1  |     17.5279 |     18.4721 |        116.565 |        236.041 |         2.47214 |
|  2  |     16.9443 |     17.5279 |         111.64 |        116.565 |         1.52786 |
|  3  |     16.5836 |     16.9443 |        127.598 |         111.64 |        0.944272 |
|  4  |     16.9443 |     17.1672 |         111.64 |        109.095 |        0.583592 |
|  5  |     17.1672 |      17.305 |        109.095 |        110.203 |         0.36068 |
|  6  |      17.082 |     17.1672 |        109.436 |        109.095 |        0.222912 |
|  7  |     17.1672 |     17.2198 |        109.095 |        109.274 |        0.137767 |
|  8  |     17.1347 |     17.1672 |        109.133 |        109.095 |       0.0851449 |
-------------------------------------------------------------------------------------------

Iteration amount = 8
Received uncertainty interval length = 0.0851449
Received accuracy = 0.0425725
Function extremum: (17.1772 ; 109.106)
Program ended with exit code: 0
```

Рис.2. Скриншот консоли

Листинг программы с реализацией алгоритмов на C++

```
#include <iostream>
#include <iomanip>
#include <cmath>

double epsilon = 0.1;

double Function(double x) {
    return x*x*exp(sin(x));
}


void OptimalSearch(double &a, double &b, unsigned int
&iterations) {
    unsigned int point_amount = 2 * (b - a) / epsilon;
    double length = epsilon / 2;
    std::pair<double, double> minPoint;
    for (unsigned int i = 0; i < point_amount; i++) {
        double tempX = a + length * (i + 1);
        std::pair<double, double> tempPoint = {tempX,
Function(tempX)};
        std::cout << "\n|";
        std::cout << std::setw(5) << i+1;
        std::cout << "  |";
        std::cout << std::setw(12) << tempPoint.first;
        std::cout << "  |";
```

```
        std::cout << std::setw(13) << tempPoint.second;
        std::cout << "  |";
        if (i == 0)
            minPoint = tempPoint;
        else if (tempPoint.second < minPoint.second)
            minPoint = tempPoint;
        iterations++;
    }
    a = minPoint.first;
    b = minPoint.second;
}

void PrintLine (double x1, double x2, unsigned int i) {
    std::cout << "\n|";
    std::cout << std::setw(3) << i;
    std::cout << "  |";
    std::cout << std::setw(12) << x1;
    std::cout << " | ";
    std::cout << std::setw(12) << x2;
    std::cout << " |";
    std::cout << std::setw(15) << Function(x1);
    std::cout << " |";
    std::cout << std::setw(15) << Function(x2);
    std::cout << " |";
}
void GoldenRatioSearch(double &a, double &b, unsigned int
&iterations) {
    const double u = (1 + sqrt(5)) / 2;

    double x1 = b - (b - a) / u;
    double x2 = a + (b - a) / u;
    while (std::abs(b - a) >= epsilon) {
        if (Function(x1) <= Function(x2)) {
            PrintLine(x1, x2, iterations+1);
            b = x2;
            x2 = x1;
            x1 = b - (b - a) / u;
            std::cout << std::setw(14) << b - a << "  |";
        }
        else {
            PrintLine(x1, x2, iterations+1);
            a = x1;
            x1 = x2;
            x2 = a + (b - a) / u;
            std::cout << std::setw(14) << b - a << "  |";
        }

        iterations++;
```

```
    }
}

int main() {
    double a = 16, b = 20;
    unsigned int iterations = 0;


    std::cout << "\tOptimal Passive Search Method";
    std::cout << "\naccuracy = " << epsilon / 2;
    std::cout <<
"\n_____________________________________\n";
    std::cout << "|   №   |  Значение x  | Значение f(x)
|";
    std::cout <<
"\n-------------------------------------";

    OptimalSearch(a, b, iterations);

    std::cout <<
"\n_____________________________________\n";

    std::cout << "\nIteration amount = " << iterations;
    std::cout << "\nReceived uncertainty interval length =
" << epsilon;
    std::cout << "\nFunction extremum: (" <<
    a << " ; " << b << ")";


    a = 16;
    b = 20;
    iterations = 0;

    std::cout << "\n\n\n\tGolden Ratio Method";
    std::cout <<
"\n_____________________________________________________
_____________________________\n";
    std::cout << "|  №  | Значение x1 | Значение x2 |
Значение f(x1) | Значение f(x2) | Длина интервала |";
    std::cout <<
"\n-----------------------------------------------------
-----------------------------";

    GoldenRatioSearch(a, b, iterations);

    std::cout <<
"\n_____________________________________________________
_____________________________\n";
```

```
    std::cout << "\nIteration amount = " << iterations;
    std::cout << "\nReceived uncertainty interval length =
" << b - a;
    std::cout << "\nReceived accuracy = " << (b - a) / 2;
    std::cout << "\nFunction extremum: (" <<
    a+(b - a) / 2 << " ; " << Function(a+(b - a) / 2) <<
")";
    std::cout<<std::endl;
    return 0;
}
```

Контрольный вопрос

В чем состоит сущность метода оптимального пассивного поиска?

Минимаксный метод поиска, в котором информация о значениях функции, вычисленных в предшествующих точках, не может быть использована, называют оптимальным пассивным поиском.
Оптимальный пассивный поиск состоит в выборе точек, равномерно расположенных на отрезке. При этом дает оценку скорости сходимости пассивного поиска с ростом числа N точек, так как скорость сходимости любого метода прямого поиска можно характеризовать скоростью уменьшения интервала неопределенности с возрастанием N.

Вывод

Таким образом, в результате вычисления минимума унимодальной на данном отрезке функции различными методами, мы убедились в том, что количество итераций для метода золотого сечения меньше, чем для метода оптимального пассивного поиска, следовательно, он эффективнее.