|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования  Российской Федерации | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования | | |
| «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра теоретической и прикладной информатики | | |
|  | | |
| Лабораторная работа № 1 | | |
| по дисциплине «Методы оптимизации» | | |
|  | | |
| **Методы одномерного поиска** | | |
|  | | |
|  | Факультет: | ПМИ |
| Группа: | ПМИ-72 |
| Вариант: | 1 |
| Студент: | Сычев Егор |
|  |  |
|  |  |
| Преподаватель: | Постовалов Сергей Николаевич |
|  |  |
|  | | |
| Новосибирск | | |
| 2020 | | |

1. **Цель работы**

Ознакомиться с методами одномерного поиска [3,12], использу­емыми в многомерных методах минимизации функций переменных. Сравнить различные алгоритмы по эффективности на тестовых примерах.

1. **Задания**

Для функции :

1. Реализовать методы дихотомии, золотого сечения, исследовать их сходимость и провести сравнение по числу вычислений функции для достижения заданной точности от 10-1 до 10-7. Построить график зависимости количества вычис­лений минимизируемой функции от десятичного логарифма задаваемой точности ;
2. Реализовать алгоритм поиска интервала, содержащего минимум функции;
3. Реализовать метод Фибоначчи, сравнить его с методами дихотомии и золотого сечения.
4. **Ход работы**

**Метод дихотомии для**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | a | b | b-a |  | x1 | x2 | f(x1) | f(x2) |
| 1 | -1,5708 | 1,570796 | 3,141593 | 3,141593 | -5E-09 | 5E-09 | -5E-09 | 5E-09 |
| 2 | -1,5708 | 5E-09 | 1,570796 | 0,5 | -0,7854 | -0,7854 | -0,70711 | -0,70711 |
| 3 | -1,5708 | -0,7854 | 0,785398 | 0,5 | -1,1781 | -1,1781 | -0,92388 | -0,92388 |
| 4 | -1,5708 | -1,1781 | 0,392699 | 0,5 | -1,37445 | -1,37445 | -0,98079 | -0,98079 |
| 5 | -1,5708 | -1,37445 | 0,19635 | 0,5 | -1,47262 | -1,47262 | -0,99518 | -0,99518 |
| 6 | -1,5708 | -1,47262 | 0,098175 | 0,5 | -1,52171 | -1,52171 | -0,9988 | -0,9988 |
| 7 | -1,5708 | -1,52171 | 0,049087 | 0,5 | -1,54625 | -1,54625 | -0,9997 | -0,9997 |
| 8 | -1,5708 | -1,54625 | 0,024544 | 0,5 | -1,55852 | -1,55852 | -0,99992 | -0,99992 |
| 9 | -1,5708 | -1,55852 | 0,012272 | 0,5 | -1,56466 | -1,56466 | -0,99998 | -0,99998 |
| 10 | -1,5708 | -1,56466 | 0,006136 | 0,5 | -1,56773 | -1,56773 | -1 | -1 |
| 11 | -1,5708 | -1,56773 | 0,003068 | 0,500001 | -1,56926 | -1,56926 | -1 | -1 |
| 12 | -1,5708 | -1,56926 | 0,001534 | 0,500002 | -1,57003 | -1,57003 | -1 | -1 |
| 13 | -1,5708 | -1,57003 | 0,000767 | 0,500003 | -1,57041 | -1,57041 | -1 | -1 |
| 14 | -1,5708 | -1,57041 | 0,000384 | 0,500007 | -1,5706 | -1,5706 | -1 | -1 |
| 15 | -1,5708 | -1,5706 | 0,000192 | 0,500013 | -1,5707 | -1,5707 | -1 | -1 |
| 16 | -1,5708 | -1,5707 | 9,59E-05 | 0,500026 | -1,57075 | -1,57075 | -1 | -1 |
| 17 | -1,5708 | -1,57075 | 4,79E-05 | 0,500052 | -1,57077 | -1,57077 | -1 | -1 |
| 18 | -1,5708 | -1,57077 | 2,4E-05 | 0,500104 | -1,57078 | -1,57078 | -1 | -1 |
| 19 | -1,5708 | -1,57078 | 1,2E-05 | 0,500209 | -1,57079 | -1,57079 | -1 | -1 |
| 20 | -1,5708 | -1,57079 | 6E-06 | 0,500417 | -1,57079 | -1,57079 | -1 | -1 |
| 21 | -1,5708 | -1,57079 | 3,01E-06 | 0,500833 | -1,57079 | -1,57079 | -1 | -1 |
| 22 | -1,5708 | -1,57079 | 1,51E-06 | 0,501663 | -1,5708 | -1,5708 | -1 | -1 |
| 23 | -1,5708 | -1,5708 | 7,59E-07 | 0,503316 | -1,5708 | -1,5708 | -1 | -1 |
| 24 | -1,5708 | -1,5708 | 3,85E-07 | 0,506588 | -1,5708 | -1,5708 | -1 | -1 |
| 25 | -1,5708 | -1,5708 | 1,97E-07 | 0,513004 | -1,5708 | -1,5708 | -1 | -1 |
| 26 | -1,5708 | -1,5708 | 1,04E-07 | 0,525348 | -1,5708 | -1,5708 | -1 | -1 |
| 27 | -1,5708 | -1,5708 | 5,68E-08 | 0,54825 | -1,5708 | -1,5708 | -1 | -1 |

**Метод золотого сечения для**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | a | b | b-a |  | x1 | x2 | f(x1) | f(x2) |
| 1 | -1,5708 | 1,570796 | 3,141593 | 3,141593 | -0,37081 | 0,370815 | -0,36237 | 0,362375 |
| 2 | -1,5708 | 0,370815 | 1,941611 | 0,618034 | -0,82917 | -0,37081 | -0,73737 | -0,36237 |
| 3 | -1,5708 | -0,37081 | 1,199982 | 0,618034 | -1,11244 | -0,82917 | -0,89678 | -0,73737 |
| 4 | -1,5708 | -0,82917 | 0,741629 | 0,618034 | -1,28752 | -1,11244 | -0,96014 | -0,89678 |
| 5 | -1,5708 | -1,11244 | 0,458352 | 0,618034 | -1,39572 | -1,28752 | -0,98471 | -0,96014 |
| 6 | -1,5708 | -1,28752 | 0,283277 | 0,618034 | -1,46259 | -1,39572 | -0,99415 | -0,98471 |
| 7 | -1,5708 | -1,39572 | 0,175075 | 0,618034 | -1,50392 | -1,46259 | -0,99776 | -0,99415 |
| 8 | -1,5708 | -1,46259 | 0,108202 | 0,618034 | -1,52947 | -1,50392 | -0,99915 | -0,99776 |
| 9 | -1,5708 | -1,50392 | 0,066873 | 0,618034 | -1,54525 | -1,52947 | -0,99967 | -0,99915 |
| 10 | -1,5708 | -1,52947 | 0,04133 | 0,618034 | -1,55501 | -1,54525 | -0,99988 | -0,99967 |
| 11 | -1,5708 | -1,54525 | 0,025543 | 0,618034 | -1,56104 | -1,55501 | -0,99995 | -0,99988 |
| 12 | -1,5708 | -1,55501 | 0,015786 | 0,618034 | -1,56477 | -1,56104 | -0,99998 | -0,99995 |
| 13 | -1,5708 | -1,56104 | 0,009757 | 0,618034 | -1,56707 | -1,56477 | -0,99999 | -0,99998 |
| 14 | -1,5708 | -1,56477 | 0,00603 | 0,618034 | -1,56849 | -1,56707 | -1 | -0,99999 |
| 15 | -1,5708 | -1,56707 | 0,003727 | 0,618034 | -1,56937 | -1,56849 | -1 | -1 |
| 16 | -1,5708 | -1,56849 | 0,002303 | 0,618034 | -1,56992 | -1,56937 | -1 | -1 |
| 17 | -1,5708 | -1,56937 | 0,001423 | 0,618034 | -1,57025 | -1,56992 | -1 | -1 |
| 18 | -1,5708 | -1,56992 | 0,00088 | 0,618034 | -1,57046 | -1,57025 | -1 | -1 |
| 19 | -1,5708 | -1,57025 | 0,000544 | 0,618034 | -1,57059 | -1,57046 | -1 | -1 |
| 20 | -1,5708 | -1,57046 | 0,000336 | 0,618034 | -1,57067 | -1,57059 | -1 | -1 |
| 21 | -1,5708 | -1,57059 | 0,000208 | 0,618034 | -1,57072 | -1,57067 | -1 | -1 |
| 22 | -1,5708 | -1,57067 | 0,000128 | 0,618034 | -1,57075 | -1,57072 | -1 | -1 |
| 23 | -1,5708 | -1,57072 | 7,93E-05 | 0,618034 | -1,57077 | -1,57075 | -1 | -1 |
| 24 | -1,5708 | -1,57075 | 4,9E-05 | 0,618034 | -1,57078 | -1,57077 | -1 | -1 |
| 25 | -1,5708 | -1,57077 | 3,03E-05 | 0,618034 | -1,57078 | -1,57078 | -1 | -1 |
| 26 | -1,5708 | -1,57078 | 1,87E-05 | 0,618034 | -1,57079 | -1,57078 | -1 | -1 |
| 27 | -1,5708 | -1,57078 | 1,16E-05 | 0,618034 | -1,57079 | -1,57079 | -1 | -1 |
| 28 | -1,5708 | -1,57079 | 7,15E-06 | 0,618034 | -1,57079 | -1,57079 | -1 | -1 |
| 29 | -1,5708 | -1,57079 | 4,42E-06 | 0,618034 | -1,57079 | -1,57079 | -1 | -1 |
| 30 | -1,5708 | -1,57079 | 2,73E-06 | 0,618034 | -1,5708 | -1,57079 | -1 | -1 |
| 31 | -1,5708 | -1,57079 | 1,69E-06 | 0,618034 | -1,5708 | -1,5708 | -1 | -1 |
| 32 | -1,5708 | -1,5708 | 1,04E-06 | 0,618034 | -1,5708 | -1,5708 | -1 | -1 |
| 33 | -1,5708 | -1,5708 | 6,45E-07 | 0,618034 | -1,5708 | -1,5708 | -1 | -1 |
| 34 | -1,5708 | -1,5708 | 3,99E-07 | 0,618034 | -1,5708 | -1,5708 | -1 | -1 |
| 35 | -1,5708 | -1,5708 | 2,46E-07 | 0,618034 | -1,5708 | -1,5708 | -1 | -1 |
| 36 | -1,5708 | -1,5708 | 1,52E-07 | 0,618034 | -1,5708 | -1,5708 | -1 | -1 |
| 37 | -1,5708 | -1,5708 | 9,41E-08 | 0,618034 | -1,5708 | -1,5708 | -1 | -1 |

**Метод Фибоначчи для**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | a | b | b-a |  | x1 | x2 | f(x1) | f(x2) |
| 1 | -1,5708 | 1,570796 | 3,141593 | 3,141593 | -0,37081 | 0,370815 | -0,36237 | 0,362375 |
| 2 | -1,5708 | 0,370815 | 1,941611 | 0,618034 | -0,82917 | -0,37081 | -0,73737 | -0,36237 |
| 3 | -1,5708 | -0,37081 | 1,199982 | 0,618034 | -1,11244 | -0,82917 | -0,89678 | -0,73737 |
| 4 | -1,5708 | -0,82917 | 0,741629 | 0,618034 | -1,28752 | -1,11244 | -0,96014 | -0,89678 |
| 5 | -1,5708 | -1,11244 | 0,458352 | 0,618034 | -1,39572 | -1,28752 | -0,98471 | -0,96014 |
| 6 | -1,5708 | -1,28752 | 0,283277 | 0,618034 | -1,46259 | -1,39572 | -0,99415 | -0,98471 |
| 7 | -1,5708 | -1,39572 | 0,175075 | 0,618034 | -1,50392 | -1,46259 | -0,99776 | -0,99415 |
| 8 | -1,5708 | -1,46259 | 0,108202 | 0,618034 | -1,52947 | -1,50392 | -0,99915 | -0,99776 |
| 9 | -1,5708 | -1,50392 | 0,066873 | 0,618034 | -1,54525 | -1,52947 | -0,99967 | -0,99915 |
| 10 | -1,5708 | -1,52947 | 0,04133 | 0,618034 | -1,55501 | -1,54525 | -0,99988 | -0,99967 |
| 11 | -1,5708 | -1,54525 | 0,025543 | 0,618034 | -1,56104 | -1,55501 | -0,99995 | -0,99988 |
| 12 | -1,5708 | -1,55501 | 0,015786 | 0,618034 | -1,56477 | -1,56104 | -0,99998 | -0,99995 |
| 13 | -1,5708 | -1,56104 | 0,009757 | 0,618034 | -1,56707 | -1,56477 | -0,99999 | -0,99998 |
| 14 | -1,5708 | -1,56477 | 0,00603 | 0,618034 | -1,56849 | -1,56707 | -1 | -0,99999 |
| 15 | -1,5708 | -1,56707 | 0,003727 | 0,618034 | -1,56937 | -1,56849 | -1 | -1 |
| 16 | -1,5708 | -1,56849 | 0,002303 | 0,618034 | -1,56992 | -1,56937 | -1 | -1 |
| 17 | -1,5708 | -1,56937 | 0,001423 | 0,618034 | -1,57025 | -1,56992 | -1 | -1 |
| 18 | -1,5708 | -1,56992 | 0,00088 | 0,618034 | -1,57046 | -1,57025 | -1 | -1 |
| 19 | -1,5708 | -1,57025 | 0,000544 | 0,618034 | -1,57059 | -1,57046 | -1 | -1 |
| 20 | -1,5708 | -1,57046 | 0,000336 | 0,618034 | -1,57067 | -1,57059 | -1 | -1 |
| 21 | -1,5708 | -1,57059 | 0,000208 | 0,618034 | -1,57072 | -1,57067 | -1 | -1 |
| 22 | -1,5708 | -1,57067 | 0,000128 | 0,618034 | -1,57075 | -1,57072 | -1 | -1 |
| 23 | -1,5708 | -1,57072 | 7,93E-05 | 0,618034 | -1,57077 | -1,57075 | -1 | -1 |
| 24 | -1,5708 | -1,57075 | 4,9E-05 | 0,618034 | -1,57078 | -1,57077 | -1 | -1 |
| 25 | -1,5708 | -1,57077 | 3,03E-05 | 0,618034 | -1,57078 | -1,57078 | -1 | -1 |
| 26 | -1,5708 | -1,57078 | 1,87E-05 | 0,618033 | -1,57079 | -1,57078 | -1 | -1 |
| 27 | -1,5708 | -1,57078 | 1,16E-05 | 0,618037 | -1,57079 | -1,57079 | -1 | -1 |
| 28 | -1,5708 | -1,57079 | 7,15E-06 | 0,618026 | -1,57079 | -1,57079 | -1 | -1 |
| 29 | -1,5708 | -1,57079 | 4,42E-06 | 0,618056 | -1,57079 | -1,57079 | -1 | -1 |
| 30 | -1,5708 | -1,57079 | 2,73E-06 | 0,617978 | -1,5708 | -1,57079 | -1 | -1 |
| 31 | -1,5708 | -1,57079 | 1,69E-06 | 0,618182 | -1,5708 | -1,5708 | -1 | -1 |
| 32 | -1,5708 | -1,5708 | 1,04E-06 | 0,617647 | -1,5708 | -1,5708 | -1 | -1 |
| 33 | -1,5708 | -1,5708 | 6,46E-07 | 0,619048 | -1,5708 | -1,5708 | -1 | -1 |
| 34 | -1,5708 | -1,5708 | 3,97E-07 | 0,615385 | -1,5708 | -1,5708 | -1 | -1 |
| 35 | -1,5708 | -1,5708 | 2,48E-07 | 0,625 | -1,5708 | -1,5708 | -1 | -1 |
| 36 | -1,5708 | -1,5708 | 1,49E-07 | 0,6 | -1,5708 | -1,5708 | -1 | -1 |
| 37 | -1,5708 | -1,5708 | 9,93E-08 | 0,666667 | -1,5708 | -1,5708 | -1 | -1 |

**Поиск начального интервала, содержащего минимум функции**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| i | x | f(x) | x-h | x+h |
| 1 | 0 | 0 | -1E-08 | 1E-08 |
| 2 | -1E-08 | -1E-08 | 0 | -2E-08 |
| 3 | -3E-08 | -3E-08 | -1E-08 | -5E-08 |
| 4 | -7E-08 | -7E-08 | -3E-08 | -1,1E-07 |
| 5 | -1,5E-07 | -1,5E-07 | -7E-08 | -2,3E-07 |
| 6 | -3,1E-07 | -3,1E-07 | -1,5E-07 | -4,7E-07 |
| 7 | -6,3E-07 | -6,3E-07 | -3,1E-07 | -9,5E-07 |
| 8 | -1,3E-06 | -1,3E-06 | -6,3E-07 | -1,9E-06 |
| 9 | -2,6E-06 | -2,5E-06 | -1,3E-06 | -3,8E-06 |
| 10 | -5,1E-06 | -5,1E-06 | -2,6E-06 | -7,7E-06 |
| 11 | -1E-05 | -1E-05 | -5,1E-06 | -1,5E-05 |
| 12 | -2E-05 | -2E-05 | -1E-05 | -3,1E-05 |
| 13 | -4,1E-05 | -4,1E-05 | -2E-05 | -6,1E-05 |
| 14 | -8,2E-05 | -8,2E-05 | -4,1E-05 | -0,00012 |
| 15 | -0,00016 | -0,00016 | -8,2E-05 | -0,00025 |
| 16 | -0,00033 | -0,00033 | -0,00016 | -0,00049 |
| 17 | -0,00066 | -0,00066 | -0,00033 | -0,00098 |
| 18 | -0,00131 | -0,00131 | -0,00066 | -0,00197 |
| 19 | -0,00262 | -0,00262 | -0,00131 | -0,00393 |
| 20 | -0,00524 | -0,00524 | -0,00262 | -0,00786 |
| 21 | -0,01049 | -0,01049 | -0,00524 | -0,01573 |
| 22 | -0,02097 | -0,02097 | -0,01049 | -0,03146 |
| 23 | -0,04194 | -0,04193 | -0,02097 | -0,06291 |
| 24 | -0,08389 | -0,08379 | -0,04194 | -0,12583 |
| 25 | -0,16777 | -0,16699 | -0,08389 | -0,25166 |
| 26 | -0,33554 | -0,32928 | -0,16777 | -0,50332 |
| 27 | -0,67109 | -0,62184 | -0,33554 | -1,00663 |
| 28 | -1,34218 | -0,97398 | -0,67109 | -2,01327 |

1. **Текст программы**

**one\_dimension\_searchers.rs**

use std::fs;

pub fn dichotomy(

f: &dyn Fn(f64) -> f64,

mut a: f64,

mut b: f64,

eps: f64,

delta: f64,

) -> (f64, i32, i32, String) {

let mut iters = 0;

let mut result = String::new();

while b - a > eps {

iters += 1;

let x1 = (a + b - delta) / 2f64;

let x2 = (a + b + delta) / 2f64;

result += &format!("{};{};;;{};{};{};{};\n", a, b, x1, x2, f(x1), f(x2));

if f(x1) < f(x2) {

b = x2;

} else {

a = x1;

}

}

let x = (a + b) / 2f64;

result += &format!("{};{};;;{};{};{};{};\n", a, b, x, x, f(x), f(x));

(x, iters \* 2, iters, result)

}

pub fn golden\_ratio(

f: &dyn Fn(f64) -> f64,

mut a: f64,

mut b: f64,

eps: f64,

) -> (f64, i32, i32, String) {

let mut x1 = a + (3f64 - 5f64.sqrt()) / 2f64 \* (b - a);

let mut x2 = b + (5f64.sqrt() - 3f64) / 2f64 \* (b - a);

let mut f1 = f(x1);

let mut f2 = f(x2);

let mut iters = 0;

let mut result = String::new();

result += &format!("{};{};;;{};{};{};{};\n", a, b, x1, x2, f1, f2);

while b - a > eps {

iters += 1;

if f1 < f2 {

b = x2;

x2 = x1;

x1 = a + (3f64 - 5f64.sqrt()) / 2f64 \* (b - a);

f2 = f1;

f1 = f(x1);

} else {

a = x1;

x1 = x2;

x2 = b + (5f64.sqrt() - 3f64) / 2f64 \* (b - a);

f1 = f2;

f2 = f(x2);

}

result += &format!("{};{};;;{};{};{};{};\n", a, b, x1, x2, f1, f2);

}

((a + b) / 2f64, iters + 1, iters, result)

}

pub fn fibbonaci(

f: &dyn Fn(f64) -> f64,

mut a: f64,

mut b: f64,

eps: f64,

) -> (f64, i32, i32, String) {

let mut fn1 = 1u128;

let mut fn2 = 1u128;

let mut iters = 0;

let mut result = String::new();

while fn2 as f64 <= (b - a) / eps {

let ft = fn2;

fn2 += fn1;

fn1 = ft;

}

let mut x1 = a + ((fn2 - fn1) as f64) / (fn2 as f64) \* (b - a);

let mut x2 = a + (fn1 as f64) / (fn2 as f64) \* (b - a);

let mut f1 = f(x1);

let mut f2 = f(x2);

result += &format!("{};{};;;{};{};{};{};\n", a, b, x1, x2, f1, f2);

while b - a > eps {

iters += 1;

if f2 < f2 {

b = x2;

x2 = x1;

x1 = a + ((fn2 - fn1) as f64) / (fn2 as f64) \* (b - a);

f2 = f1;

f1 = f(x1);

} else {

a = x1;

x1 = x2;

x2 = a + (fn1 as f64) / (fn2 as f64) \* (b - a);

f1 = f2;

f2 = f(x2);

}

let ft = fn1;

fn1 = fn2 - fn1;

fn2 = ft;

result += &format!("{};{};;;{};{};{};{};\n", a, b, x1, x2, f1, f2);

}

((a + b) / 2f64, iters + 1, iters, result)

}

pub fn get\_interval(f: &dyn Fn(f64) -> f64, mut x: f64, delta: f64) -> (f64, f64, String) {

let mut h = delta;

let mut result = String::new();

result += &format!("{};{};{};{};\n", x, f(x), x - delta, x + delta);

if f(x) > f(x + delta) {

x += delta;

} else if f(x) > f(x - delta) {

x -= delta;

h = -delta;

}

while f(x) > f(x + h) {

result += &format!("{};{};{};{};\n", x, f(x), x - h, x + h);

h \*= 2f64;

x += h;

}

result += &format!("{};{};{};{};\n", x, f(x), x - h, x + h);

(x - h, x + h, result)

}

pub fn run(f: &dyn Fn(f64) -> f64, a: f64, b: f64, eps: f64) -> (i32, i32, i32) {

// find interval

let interval = get\_interval(f, (a + b) / 2f64, eps / 10f64);

fs::write("./lab1/interval.csv", interval.2);

// run calculations

let dichotomy\_result = dichotomy(&f, a, b, eps, eps / 10f64);

let golden\_ratio\_result = golden\_ratio(&f, a, b, eps);

let fibbonaci\_result = fibbonaci(&f, a, b, eps);

// results

fs::write("./lab1/fibbonaci.csv", fibbonaci\_result.3);

fs::write("./lab1/golden\_ratio.csv", golden\_ratio\_result.3);

fs::write("./lab1/dichotomy.csv", dichotomy\_result.3);

(

dichotomy\_result.1,

golden\_ratio\_result.1,

fibbonaci\_result.1,

)

}

**main.rs**

use std::fs;

pub mod one\_dimension\_searchers;

fn f(x: f64) -> f64 {

    x.sin()

}

fn main() {

    let mut result = String::new();

    (1..=7).for\_each(|e| {

        let func\_calls = one\_dimension\_searchers::run(

            &f,

            -std::f64::consts::FRAC\_PI\_2,

            std::f64::consts::FRAC\_PI\_2,

            10f64.powf(-e as f64),

        );

        result += &format!("{};{};{};\n", func\_calls.0, func\_calls.1, func\_calls.2);

    });

    fs::write("./lab1/precisions.csv", result);

}

1. **Выводы**

Из рассмотренных методов, метод дихотомии является самым простым и находит решение за меньшее число итераций. Однако, в методах золотого сечения и Фибоначчи, за каждую итерацию новое значение функции вычисляется только один раз, что повысит скорость этих методов по отношению к методу дихотомии, если рассматриваемая функция будет сложнее для вычисления.

Метод Фибоначчи, по количеству итераций и вычислений функции, идентичен методу золотого сечения, но требует дополнительно вычислить n+2 число Фибоначчи. Это требует дополнительных затрат, но позволяет оценить количество итераций для достижения необходимой точности. Так же, при большой требуемой точности, n+2 число Фибоначчи становятся слишком большим, из-за чего ускоряется накопление погрешности.