МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

# «КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ» (ФГБОУ ВО «КубГУ»)

**Факультет компьютерных технологий и прикладной математики Кафедра прикладной математики**

# ОТЧЕТ

**по дисциплине**

# «Методы оптимизации»

Работу выполнил М. А. Пузырёв

Работу принял преподаватель Е. С. Троценко

Краснодар 2024

# Постановка задачи

Требуется найти безусловный минимум функции 𝑓(𝑥) т.е. такую точку 𝑥∗ ∈ 𝑅, что 𝑓(𝑥∗) = min 𝑓(𝑥), x *R*. Мы будем искать безусловный минимум функции 𝑓(𝑥) = x2 + 2 методом Фибоначчи на промежутке [−3; 7], с заданными 𝜀 = 0,2 и 𝑙 = 0,5.

# Стратегия поиска

Метод относится к последовательным стратегиям. Задаётся начальный интервал неопределённости и количество N вычислений функции. Алгоритм уменьшения интервала опирается на анализ значений функции в двух точках. Точки вычисления функции находятся с использованием последовательности из N+1 чисел Фибоначчи. Как в методе золотого сечения, на первой итерации требуются два вычисления функции, а на каждой последующей – только по одному. Условия окончания стандартные: поиск заканчивается, когда длина текущего интервала неопределённости оказывается меньше установленной длины.

# Алгоритм

Шаг 1. Задать начальный интервал неопределённости 𝐿0 = [𝑎0, 𝑏0],

𝑙>0 – допустимую длину конечного интервала, > 0 – константу различимости.

Шаг 2. Найти количество N вычислений функции как наименьшее целое число, при котором удовлетворяется условие FN  , и числа Фибоначчи F0, F1,…,FN.

Шаг 3. Положить 𝑘 = 0.

Шаг 4. Вычислить y0 = a0 + b0 – a0); z0 = a0 + (b0 – a0).

Шаг 5. Вычислить 𝑓(yk), 𝑓(zk).

Шаг 6. Сравнить 𝑓(yk) и 𝑓(zk):

а) если 𝑓(yk) 𝑓(zk), положить

ak+1 = ak, bk+1 = zk; zk+1 = yk; yk+1 = ak+1 + (bk+1 – ak+1).

б) если 𝑓(yk) 𝑓(zk), положить

ak+1 = yk, bk+1 = bk; yk+1 = zk; zk+1 = ak+1 + (bk+1 – ak+1).

Перейти к шагу 7.

Шаг 7. Проверить условие окончания и в случае необходимости сделать заключительное N-е вычисление функции для получения решения:

а) если k N – 3, положить k = k+1 и перейти к шагу 5;

б) если k = N – 3, то всегда yN-2 = zN-2 = , т.е. отсутствует точка нового вычисления функции. Следует положить: yN-1 = yN-2 = zN-2; zN-1 = yN-1 + .

В точках yN-1 и zN-1 вычисляются значения функции и находятся границы конечного интервала неопределённости:

- если 𝑓(yN-1) 𝑓(zN-1), положить aN-1 = aN-2, bN-1 = zN-1;

- если (yN-1) 𝑓(zN-1), положить aN-1 = yN-1, bN-1 = bN-2.

Процесс поиска завершается и x\* [aN-1, bN-1]. В качестве приближённого решения можно взять любую точку последнего интервала, например, его середину x\* = .

# Код программы

# Реализация кода представлена на языке C++:

#include <iostream>

#include <cmath>

#include <math.h>

using namespace std;

double F(double x) {

return pow(x, 2) + 2;

}

int main() {

setlocale(LC\_ALL, "rus");

int it = 0, i = 1;

double a = -3.0;

double b = 7.0;

double l = 0.5;

double eps = 0.2;

double L = abs(a - b) / (2 \* l);

double fib[20], f1;

fib[0] = 1, fib[1] = 1;

do {

i++;

fib[i] = fib[i - 1] + fib[i - 2];

} while (fib[i] <= L && i <= 19);

double y = a + fib[i - 2] \* (b - a) / fib[i];

double z = a + fib[i - 1] \* (b - a) / fib[i];

do {

if (F(y) <= F(z)) {

b = z;

z = y;

y = a + fib[i - it - 3]\*(b - a) / fib[i - it - 1];

}

else {

a = y;

y = z;

z = a + fib[i - it - 2] \* (b - a) / fib[i - it - 1];

}

it++;

} while (it <= i-3);

y = (a + b) / 2;

z = y + eps;

if (F(y) <= F(z))

b = z;

else

a = y;

double min = (a + b) / 2;

cout << "Алгоритм сошёлся за " << it << " итераций." << endl;

cout << "Искомый минимум функции: " << min << endl;

cout << "Значение в минимуме: " << F(min) << endl;

cout << "Сходимость заданной функции: " << 1 / fib[i] << endl;

return 0;

}

# Сходимость

Для метода Фибоначчи характеристика относительного уменьшения начального интервала неопределенности находится по формуле R(*N*) = ,

где *N* – количество вычислений функции.

# Вывод

Для решения задачи по поиску безусловного минимума функции

𝑓(x) = x2 + 2\*x – методом Фибоначчи была написана программа на языке C++. Результатом работы программы является вычисленное минимальное значение функции, равное 2,04314 в точке минимума, равной -0,207692, при этом сходимость равна 0,0769231 (рис.1). При изменении допустимой длины конечного интервала 𝑙 точка минимума, значение функции в ней и сходимость меняются. При увеличении или уменьшении 𝑙 соответственно увеличивается или уменьшается количество итераций алгоритма, необходимых для нахождения безусловного минимума, и как следствие сходимость уменьшается.

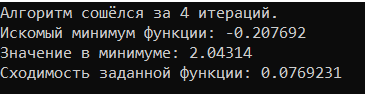


Рисунок 1 – результат работы программы