Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 1

Дисциплина Методы вычислений

Тема Метод поразрядного поиска

Вариант №10

Студент Коноваленко В. Д.

Группа ИУ7-21М

Оценка (баллы) _______________

Преподаватель Власов П.А.

Москва.
2024 г.

Цель работы: изучение метода поразрядного поиска для решения задачи одномерной минимизации.

Содержание работы

1. реализовать метод поразрядного поиска в виде программы на ЭВМ;
2. провести решение задачи
$$\begin{cases} f(x) \to min \\ x \in [a, b] \end{cases}$$
для данных индивидуального варианта;
3. организовать вывод на экран графика целевой функции, найденной точки минимума $(x^*,\ f(x^*))$ и последовательности точек $\big(x_i, f(x_i)\big)$, приближающих точку искомого минимума (для последовательности точек следует предусмотреть возможность «отключения» вывода её на экран).

Целевая функция *f(x)*	***[a, b]***
$\sin\left(\dfrac{x^4 + x^3 - 3x + 3 - 30^{\frac{1}{3}}}{2}\right) + th\left(\dfrac{4\sqrt{3}x^3 - 2x - 6\sqrt{2} + 1}{-2\sqrt{3}x^3 + x + 3\sqrt{2}}\right) + 1.2$	*[0, 1]*

Метод поразрядного поиска является усовершенствованием метода перебора с целью уменьшения числа обращений к целевой функции.

Основная идея: на начальном этапе, используя сравнительно большой шаг определяют примерную локализацию точки минимума, далее в полученной окрестности значение точки минимума уточняют с использованием более мелкого шага.

В основе метода лежит одно из свойств унимодальных функций:

Если $x_1 < x_2$, то

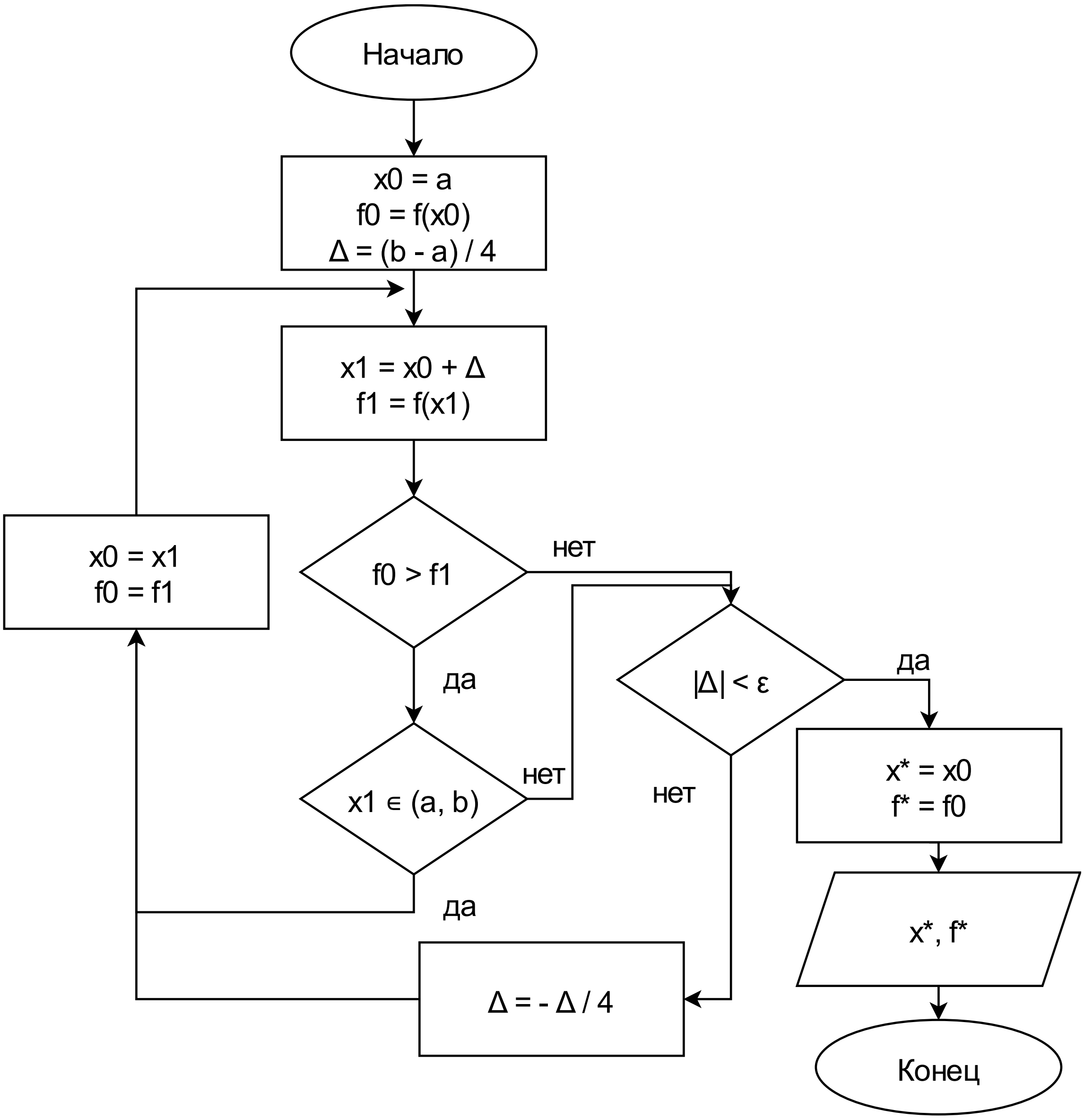
$f(x_1) \leq f(x_2)$ => $x^* \in [a;\ x_2]$

$f(x_1) > f(x_2)$ => $x^* \in [x_1;\ b]$

Обычно начальное значение шага берут $\Delta = \frac{b-a}{4}$, затем вычисляют значения

$$f(x_i) = f(a + i\Delta), i = 0, 1, 2 \ldots$$

до тех пор, пока на некотором шаге не будет выполнено условие: $f(x_i) < f(x_{i+1})$. В этих случаях направление поиска изменяют на противоположное и уменьшают шаг (как правило, также в 4 раза).



Текст программы представлен на Листинге 1

Листинг 1

```
function lab1()
    debug = 1;

    a = 0;
    b = 1;
    e = 0.01;

    [x, y, N] = bitwiseSearch(a, b, e, debug);

    fplot(@f, [a, b]);
    hold on;
```

```
    fprintf('RESULT: e = %f | N = %d | x* = %.10f | f(x*) = %.10f', e, N, x, y);
    scatter(x, y, 'b', 'filled');
end

function y = f(x)
    y = sin((power(x, 4) + power(x, 3) - 3 * x + 3 - power(30, 1/3)) / 2) + tanh((4
* sqrt(3) * power(x, 3) - 2 * x - 6 * sqrt(2) + 1) / (-2 * sqrt(3) * power(x, 3) +
x + 3 * sqrt(2))) + 1.2;
end

function [x, y, N] = bitwiseSearch(a, b, e, debug)
    i = 0;
    x0 = a;
    f0 = f(x0);
    delta = (b - a) / 4;

    if debug
        hold on;
        fprintf('%d: x%d = %.10f | y%d = %.10f\n', 0, 0, x0, 0, f0)
        scatter(x0, f0);
    end

    while 1
        i = i + 1;
        x1 = x0 + delta;
        f1 = f(x1);

        if debug
            fprintf('%d: x%d = %.10f | y%d = %.10f\n', i, i, x1, i, f1)
            scatter(x1, f1);
        end

        if f0 <= f1 || x1 <= a || x1 >= b
            if abs(delta) < e
                break;
            else
                delta = -delta / 4;
            end
        end
        x0 = x1;
        f0 = f1;
    end

    x = x0;
    y = f0;
    N = i + 1;

end
```

Результаты расчетов для задачи из индивидуального варианта.

№ п/п	ε	N	x^*	$f(x^*)$
1	0.01	21	0.7070312500	-0.4652470295
2	0.0001	30	0.7054443359	-0.4652516055
3	0.000001	47	0.7054662704	-0.4652516064