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 (ИУ)»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии (ИУ7)»

ОТЧЕТ

Лабораторная работа №1

по курсу «Методы вычислений»

на тему: «Метод поразрядного поиска»

Вариант № 6

Студент ИУ7-22М (Группа) _____ (Подпись, дата) К.Э. Ковалец (И. О. Фамилия)

Преподаватель _____ (Подпись, дата) П.А. Власов (И. О. Фамилия)

2024 г.

1 Теоретическая часть

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

Задание:

1. Реализовать метод поразрядного поиска в виде программы на ЭВМ.

2. Провести решение задачи:

$$\begin{cases} f(x) \to min, \\ x \in [a, b] \end{cases} \tag{1.1}$$

   для данных индивидуального варианта;

3. организовать вывод на экран графика целевой функции, найденной точки минимума $(x^*,\ f(x^*))$ и последовательности точек $(x_i,\ f(x_i))$, приближающих точку искомого минимума (для последовательности точек следует предусмотреть возможность «отключения» вывода ее на экран).

Таблица 1.1 – Данные индивидуального варианта

№ вар.	**Целевая функция** $f(x)$	**[a, b]**
6	$\text{ch}\ (\frac{3x^3\ +\ 2x^2\ -\ 4x\ +\ 5}{3}) + \text{th}\ (\frac{x^3\ -\ 3\sqrt{2}x\ -\ 2}{2x\ +\ \sqrt{2}})\ -\ 2.5$	[0, 1]

1.1 Краткое описание метода поразрядного поиска

Метод поразрядного поиска является усовершенствованием метода перебора для 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 (как правило, уменьшенного в 4 раза).

В основе метода лежит известное свойство унимодальных функци: если

$x_1 < x_2$, то

$$f(x_1) \leqslant f(x_2) \Rightarrow x^* \in [a,\ x_2],$$
$$f(x_1) > f(x_2) \Rightarrow x^* \in [x_1,\ b]. \tag{1.2}$$

Схема рассматриваемого метода представлена на рисунке 1.1.

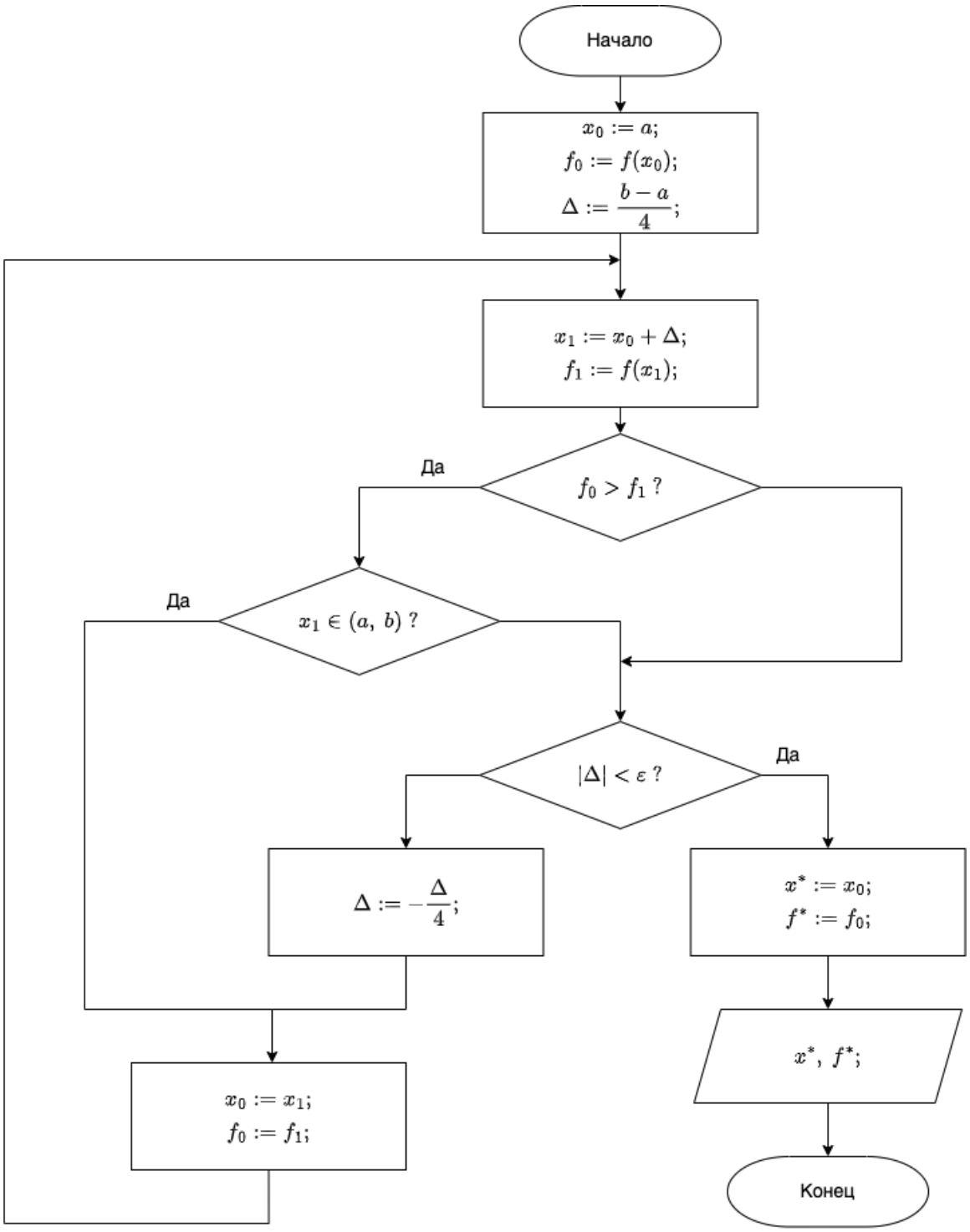

Рисунок 1.1 – Схема алгорима метода поразрядного поиска

2 Практическая часть

Таблица 2.1 – Результаты расчетов для задачи из индивидуального варианта

№ п/п	ε	N	x^*	$f(x^*)$
1	0.01	19	0.4804687500	-1.4738794316
2	0.0001	36	0.4824218750	-1.4738932843
3	0.000001	50	0.4824180603	-1.4738932844

В листинге 2.1 представлен код программы.

Листинг 2.1 — Код программы

```
function lab_01()
    clc();

    debugFlg = true;
    delay = 0;
    a = 0;
    b = 1;
    eps = 0.01;

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

    [xStar, fStar] = bitwiseSearch(a, b, eps, debugFlg, delay);
    scatter(xStar, fStar, 'r', 'filled');
end

function [x0, f0] = bitwiseSearch(a, b, eps, debugFlg, delay)
    i = 1;
    x0 = a;
    f0 = f(x0);
    delta = (b - a) / 4;

    if debugFlg
        fprintf('N = %2d:   x0 = %.10f;   f(x0) = %.10f;\n', i, x0, f0);
    end

    plot_x = [];
    plot_f = [];
    while true
        i = i + 1;
        x1 = x0 + delta;
        f1 = f(x1);
```

Продолжение листинга 2.1

```
33
34          if debugFlg
35              fprintf('N = %2d:   x1 = %.10f;   f(x1) = %.10f;\n', i, x1, f1);
36
37              plot_x(end + 1) = x1;
38              plot_f(end + 1) = f1;
39
40              plot(plot_x, plot_f, '*k');
41              plot(x1, f1, '*r');
42
43              pause(delay);
44          end
45
46          if f0 > f1
47              if x1 <= a || x1 >= b
48                  if abs(delta) < eps
49                      break;
50                  else
51                      delta = - delta / 4;
52                  end
53              end
54
55              x0 = x1;
56              f0 = f1;
57          else
58              if abs(delta) < eps
59                  break;
60              else
61                  delta = - delta / 4;
62              end
63
64              x0 = x1;
65              f0 = f1;
66          end
67      end
68
69      fprintf('\nОтвет:   x* = %.10f;   f(x*) = %.10f.\n', x0, f0);
70
71      if debugFlg
72          plot(plot_x, plot_f, '*k');
73      end
74  end
```