



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:

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

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	$\text{ch} \left(\frac{3x^3 + 2x^2 - 4x + 5}{3} \right) + \text{th} \left(\frac{x^3 - 3\sqrt{2}x - 2}{2x + \sqrt{2}} \right) - 2.5$	[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 функций: если

$x_1 < x_2$, то

$$\begin{aligned} f(x_1) \leq f(x_2) &\Rightarrow x^* \in [a, x_2], \\ f(x_1) > f(x_2) &\Rightarrow x^* \in [x_1, b]. \end{aligned} \quad (1.2)$$

Схема рассматриваемого метода представлена на рисунке 1.1.

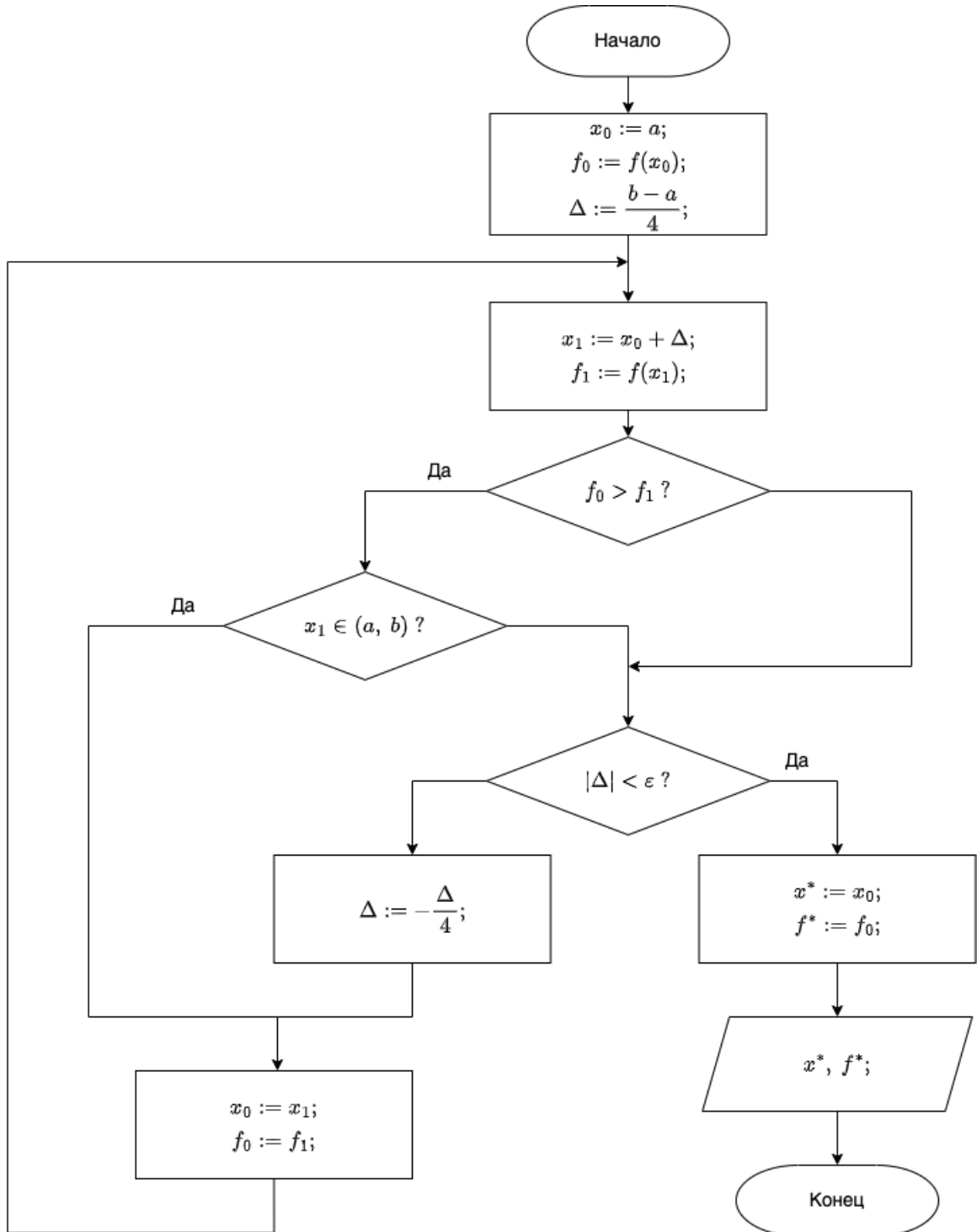


Рисунок 1.1 – Схема алгоритма метода поразрядного поиска

2 Практическая часть

Таблица 2.1 – Результаты расчетов для задачи из индивидуального варианта

№ п/п	ε	N	x^*	$f(x^*)$
1	0.01	18	0.4804687500	-1.4738794316
2	0.0001	35	0.4824218750	-1.4738932843
3	0.000001	49	0.4824180603	-1.4738932844

В листинге 2.1 представлен код программы.

Листинг 2.1 — Код программы

```
1  function lab_01()
2      clc();
3
4      debugFlg = true;
5      delay = 0.5;
6      a = 0;
7      b = 1;
8      eps = 0.02;
9
10     fplot(@f, [a, b]);
11     hold on;
12
13     [xStar, fStar] = bitwiseSearch(a, b, eps, debugFlg, delay);
14     scatter(xStar, fStar, 'r', 'filled');
15 end
16
17 function [x0, f0] = bitwiseSearch(a, b, eps, debugFlg, delay)
18     i = 0;
19     x0 = a;
20     f0 = f(x0);
21     delta = (b - a) / 4;
22
23     plot_x = [];
24     plot_f = [];
25     while true
26         i = i + 1;
27         x1 = x0 + delta;
28         f1 = f(x1);
29
30         if debugFlg
31             fprintf('Итерация %2d:   x1 = %.10f;   f(x1) = %.10f;\n', i, x1, f1);
32         end
33         x0 = x1;
34         f0 = f1;
35         delta = delta / 2;
36     end
37     [xStar, fStar] = [x0, f0];
38 end
```

Продолжение листинга 2.1

```
33         plot_x(end + 1) = x1;
34         plot_f(end + 1) = f1;
35
36         plot(plot_x, plot_f, '*k');
37         plot(x1, f1, '*r');
38
39         pause(delay);
40     end
41
42     if f0 > f1
43         if x1 <= a || x1 >= b
44             if abs(delta) < eps
45                 break;
46             else
47                 delta = - delta / 4;
48             end
49         end
50
51         x0 = x1;
52         f0 = f1;
53     else
54         if abs(delta) < eps
55             break;
56         else
57             delta = - delta / 4;
58         end
59
60         x0 = x1;
61         f0 = f1;
62     end
63 end
64
65 fprintf('\nОтвет:  x* = %.10f;   f(x*) = %.10f.\n', x0, f0);
66
67 if debugFlg
68     plot(plot_x, plot_f, '*k');
69 end
70 end
71
72 function y = f(x)
73     y = cosh((3 .* power(x, 3) + 2 .* power(x, 2) - 4 .* x + 5) ./ 3) +
↪     tanh((power(x, 3) - 3 .* power(2, 1/2) .* x - 2) ./ (2 .* x + power(2, 1/2))) -
↪     2.5;
74 end
```