



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 №2

по курсу «Методы вычислений»

на тему: «Метод золотого сечения»

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 золотого сечения для решения задачи одномерной минимизации.

Задание:

1. Реализовать метод золотого сечения 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	$\operatorname{ch} \left(\frac{3x^3 + 2x^2 - 4x + 5}{3} \right) + \operatorname{th} \left(\frac{x^3 - 3\sqrt{2}x - 2}{2x + \sqrt{2}} \right) - 2.5$	[0, 1]

1.1 Краткое описание метода золотого сечения

Пусть $x_1, x_2 \in [a, b]$ — пробные точки, которые расположены симметрично относительно середины отрезка. С целью уменьшения количества вычисляемых значений функции f надо подобрать x_1 и x_2 так, чтобы при переходе к новому отрезку $[a_1, b_1] \subset [a, b]$ одна из них стала новой пробной точкой.

1. Каждая из точек x_1 и x_2 , используемых в данном методе, делит отрезок $[a, b]$ на 2 неравные части так, что

$$\frac{\text{длина большей части отрезка}}{\text{длина всего отрезка}} = \frac{\text{длина меньшей части отрезка}}{\text{длина большей части отрезка}}. \quad (1.2)$$

Точки, обладающие этим свойством, называются точками золотого сечения отрезка $[a, b]$.

2. На каждой итерации длина отрезка уменьшается в $\tau = \frac{\sqrt{5}-1}{2}$ раз, поэтому после n итераций длина соответствующего отрезка равна $\tau^n(b-a)$.
3. Число n итераций, необходимых для достижения заданной точности, составляет

$$2.1 \cdot \ln\left(\frac{b-a}{2\varepsilon}\right). \quad (1.3)$$

4. Для выполнения первой итерации метода необходимо вычислить 2 значения функции f . Для выполнения остальных итераций необходимо вычислить 1 значение функции f . Таким образом, для выполнения n итераций необходимо вычисление $N+1$ значения функции. Точность $\varepsilon(N)$, которую можно обеспечить путем вычисления N значений функции f , равна

$$\frac{1}{2}\tau^{N-1}(b-a) \approx \tau^{N-2}(b-a). \quad (1.4)$$

Схема рассматриваемого метода представлена на рисунке 1.1.

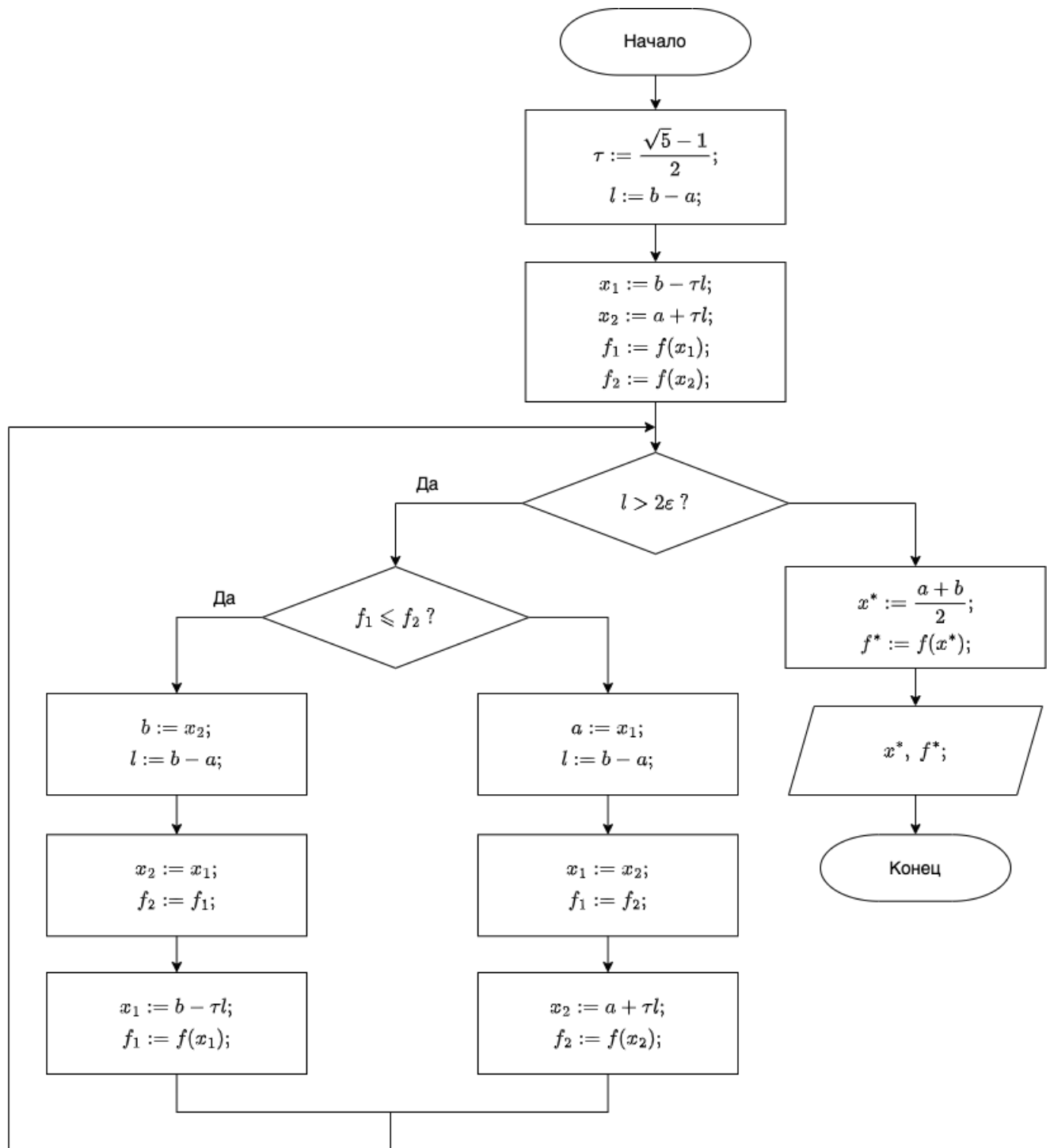


Рисунок 1.1 – Схема алгоритма метода поразрядного поиска

2 Практическая часть

Таблица 2.1 – Результаты расчетов для задачи из индивидуального варианта

№ п/п	ε	N	x^*	$f(x^*)$
1	0.01	11	0.4787137637	-1.4738433053
2	0.0001	20	0.4824455579	-1.4738932816
3	0.000001	30	0.4824184653	-1.4738932844

В листинге 2.1 представлен код программы.

Листинг 2.1 — Код программы

```
1  function lab_02()
2      clc();
3
4      a = 0;
5      b = 1;
6      eps = 1e-2;
7
8      debugFlag = true;
9      delay = 0.5;
10
11     fplot(@f, [a, b]);
12     hold on;
13
14     tau = (sqrt(5) - 1) / 2;
15     l = b - a;
16
17     x1 = b - tau * l;
18     x2 = a + tau * l;
19     f1 = f(x1);
20     f2 = f(x2);
21
22     N = 2;
23     while true
24         if debugFlag
25             fprintf('N = %2d:   a = %.10f;   b = %.10f;\n', N, a, b);
26             line([a b], [f(a) f(b)]);
27             pause(delay);
28         end
29
30         if l > 2 * eps
31             if f1 <= f2
32                 b = x2;
```

Продолжение листинга 2.1

```
33         l = b - a;
34
35         x2 = x1;
36         f2 = f1;
37
38         x1 = b - tau * l;
39         f1 = f(x1);
40     else
41         a = x1;
42         l = b - a;
43
44         x1 = x2;
45         f1 = f2;
46
47         x2 = a + tau * l;
48         f2 = f(x2);
49     end
50
51     N = N + 1;
52 else
53     xStar = (a + b) / 2;
54     fStar = f(xStar);
55
56     N = N + 1;
57     break
58 end
59 end
60
61 scatter(xStar, fStar, 'r', 'filled');
62 fprintf('\nОтвет:   x* = %.10f;   f(x*) = %.10f.\n\n', xStar, fStar);
63 end
64
65 function y = f(x)
66     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;
67 end
```