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:

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

   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	$\mathrm{ch}\,(\frac{3x^3\ +\ 2x^2\ -\ 4x\ +\ 5}{3}) + \mathrm{th}\,(\frac{x^3\ -\ 3\sqrt{2}x\ -\ 2}{2x\ +\ \sqrt{2}})\ -\ 2.5$	[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

$$\frac{\text{длина большей части отрезка}}{\text{длина всего отрезка}} = \frac{\text{длина меньшей части отрезка}}{\text{длина большей части отрезка}}. \tag{1.2}$$

   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

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

4. Для выполнения первой итерации метода необходимо вычислить 2 значения функции f. Для выполнения остальных итерации необходимо вычислить 1 значения 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

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

Схема рассматриваемого метода представлена на рисунке 1.1.

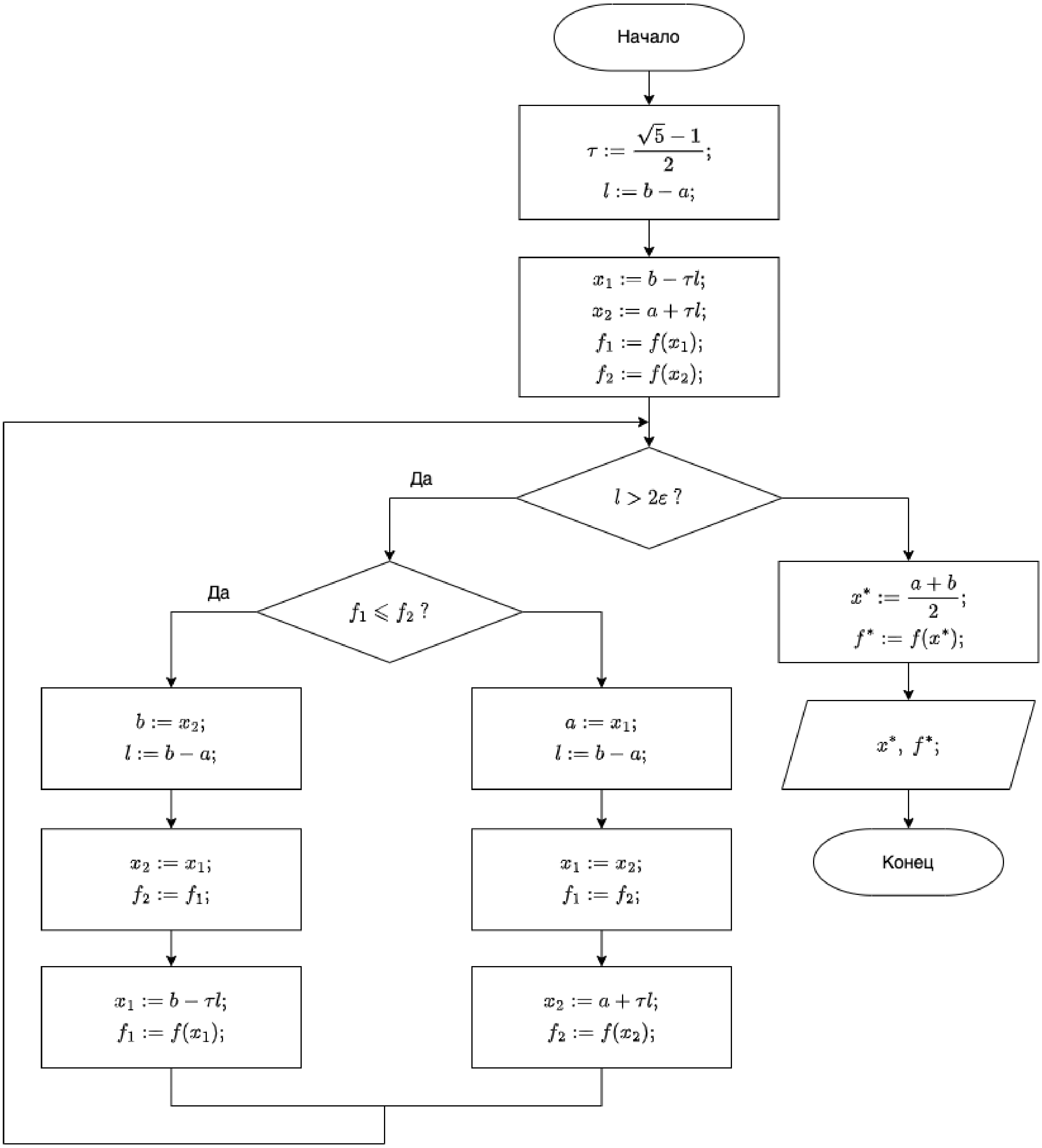

Рисунок 1.1 – Схема алгорима метода поразрядного поиска

2 Практическая часть

Таблица 2.1 – Результаты расчетов для задачи из индивидуального варианта

№ п/п	ε	N	x^*	$f(x^*)$
1	0.01	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

```
function lab_02()
    clc();

    a = 0;
    b = 1;
    eps = 1e-2;

    debugFlag = true;
    delay = 0.5;

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

    tau = (sqrt(5) - 1) / 2;
    l = b - a;

    x1 = b - tau * l;
    x2 = a + tau * l;
    f1 = f(x1);
    f2 = f(x2);

    N = 2;
    while true
        if debugFlag
            fprintf('N = %2d:    a = %.10f;    b = %.10f;\n', N, a, b);
            line([a b], [f(a) f(b)]);
            pause(delay);
        end

        if l > 2 * eps
            if f1 <= f2
                b = x2;
```

Продолжение листинга 2.1

```
                l = b - a;

                x2 = x1;
                f2 = f1;

                x1 = b - tau * l;
                f1 = f(x1);
            else
                a = x1;
                l = b - a;

                x1 = x2;
                f1 = f2;

                x2 = a + tau * l;
                f2 = f(x2);
            end

            N = N + 1;
        else
            xStar = (a + b) / 2;
            fStar= f(xStar);

            N = N + 1;
            break
        end
    end

    scatter(xStar, fStar, 'r', 'filled');
    fprintf('\nОтвет:   x* = %.10f;   f(x*) = %.10f.\n\n', xStar, fStar);
end

function y = f(x)
    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;
end
```