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»

ОТЧЕТ

по лабораторной работе №2

по курсу «Методы вычислений»

на тему: «Метод золотого сечения»

Вариант № 7

Студент ИУ7-22М (Группа) _______ (Подпись, дата) Е. О. Карпова (И. О. Фамилия)

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

2025 г.

1 Теоретический раздел

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

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

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 исходного минимума (для последовательности точек следует предусмотреть возможность «отключения» вывода ее на экран).

1.1 Исходные данные варианта №7

$$f(x) = arctg(x^3 - 5x + 1) + \left(\frac{x^2}{3x - 2}\right)^{\sqrt{3}}.$$

$$x \in [1, 2].$$

1.2 Краткое описание метода золотого сечения

Методы исключения отрезков основаны на следующих принципах.

1. Выбираем две произвольные точки x_1 и x_2 такие, что $a < x_1 < x_2 < b$.

2. Свойство унимодальной функции: если $a \leq x_1 \leq x_2 \leq b$, то

   (a) если $f(x_1) \leq f(x_2)$, то $x^* \in [a, x_2]$,

   (b) иначе $x \in [x_1, b]$.

3. Проверяем условия (a) и (b) и по результатам этой проверки отбрасываем часть отрезка $[a, b]$.

4. Вычисления продолжаются до тех пор, пока длина текущего отрезка не станет меньше ϵ — заданной точности.

Способ выбора x_1 и x_2 определяет конкретный метод поиска минимума. В методе золотого сечения для уменьшения количества значений целевой функции, которые приходится вычислять в ходе реализации алгоритма, выбирают пробные точки x_1 и x_2 внутри отрезка $[a, b]$ так, чтобы при переходе к очередному отрезку одна из этих точек стала новой пробной точкой.

При этом будем считать, что отношение длины нового отрезка к длине текущего отрезка не зависит от номера итерации и равно τ. Также, будем считать, что x_1 и x_2 располагаются симметрично относительно середины отрезка $[a, b]$.

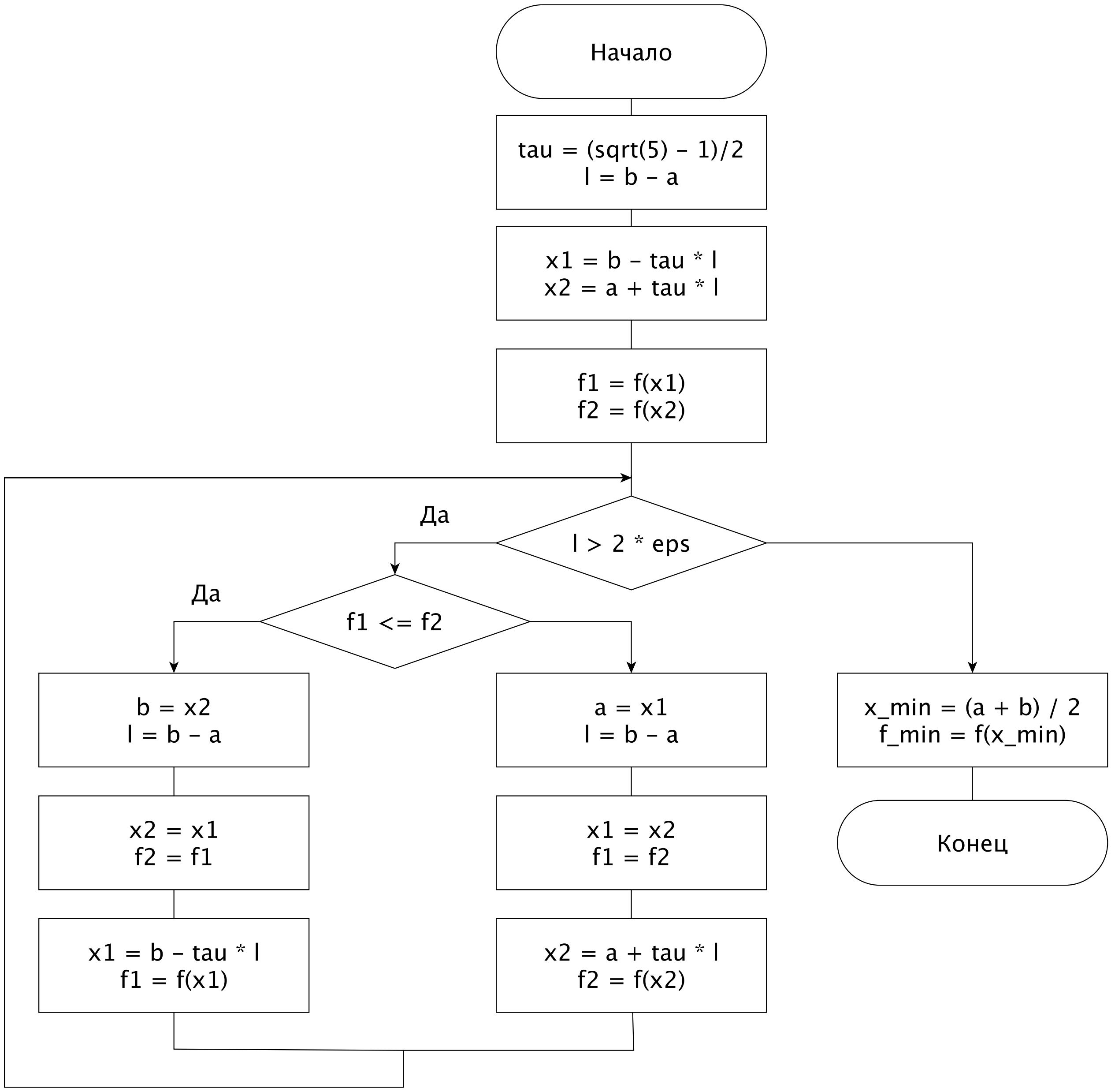

Рисунок 1.1 – Схема алгоритма исключения отрезков методом золотого сечения

2 Практический раздел

Листинг 2.1 – Исходный код программы

```
# Лабораторная работа 2. Вариант 7.

function main()
  clc;

  debug = true;

  a = 1;
  b = 2;
  eps = 1e-6;

  [x_min, f_min, n, xs, fs] = find_min(debug, a, b, eps);
  draw_plot(a, b, eps, x_min, f_min, xs, fs);
  fprintf('\n\033[36mТочка минимума (x*, f(x*)) = (%f, %f),
     количество вычислений функции: %d.\033[0m\n', x_min, f_min,
     n);
end

function [x_min, f_min, n, xs, fs] = find_min(debug, a, b, eps)
  tau = (sqrt(5) - 1) / 2;
  l = b - a;

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

  xs = [];
  fs = [];
  xs(end + 1) = x1;
  fs(end + 1) = f1;

  if debug
    fprintf('(x0, f(x0)) = (%f, %f).\n', x1, f1);
  endif

  x2 = a + tau*l;
  f2 = f(x2);
  xs(end + 1) = x2;
  fs(end + 1) = f2;
```

```
if debug
  fprintf('(x1, f(x1)) = (%f, %f).\n', x2, f2);
endif

i = 2;

while true
  if l <= 2 * eps
    x_min = (a + b) / 2;
    f_min = f(x_min);
    n = i + 1;
    return;
  endif

  if f1 <= f2
    b = x2;
    l = b - a;

    x2 = x1;
    f2 = f1;

    x1 = b - tau*l;
    f1 = f(x1);
    i = i + 1;

    xs(end + 1) = x1;
    fs(end + 1) = f1;

    if debug
      fprintf('(x%d, f(x%d)) = (%f, %f).\n', i-1, i-1, x1, f1);
    endif
  else
    a = x1;
    l = b - a;

    x1 = x2;
    f1 = f2;

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

```
        i = i + 1;

        xs(end + 1) = x2;
        fs(end + 1) = f2;

        if debug
          fprintf('(x%d, f(x%d)) = (%f, %f).\n', i-1, i-1, x2, f2);
        endif
      endif
    endwhile
end

function draw_plot(a, b, step, x_min, f_min, xs, fs)
    x = a:step:b;
    y = zeros(size(x));
    for i = 1:length(x)
        y(i) = f(x(i));
    end
    plot(x,y);
    hold on;
    for i = 1:length(xs)
        scatter(xs(i), fs(i), 8, 'g', 'filled');
    end
    scatter(x_min, f_min, 10, 'r', 'filled');
    text(x_min, f_min, sprintf('\n\n\n\n(%.3f, %.3f)', x_min,
        f_min), 'FontSize', 12);
    hold off;
end

function y = f(x)
    y = atan(x .^ 3 - 5 * x + 1) + ((x .^ 2) / (3 * x - 2)) .^
        sqrt(3);
end
```

Таблица 2.1 – Результаты расчетов по индивидуальному варианту

№	ϵ	N	x^*	$f(x^*)$
1	10^{-2}	12	1.319660	-0.460962
2	10^{-4}	21	1.321126	-0.460965
3	10^{-6}	31	1.321162	-0.460965