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 №3

по курсу «Методы вычислений»

на тему: «Метод парабол»

Вариант № 6

Студент	ИУ7-22М		К.Э. Ковалец
	(Группа)	(Подпись, дата)	(И. О. Фамилия)
Преподаватель			П.А. Власов
		(Подпись, дата)	(И. О. Фамилия)

2024 г.

1 Теоретическая часть

Цель работы: изучение метода парабол для решения задачи одномерной минимизации.

Задание:

1. Реализовать метод парабол в сочетании с методом 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 парабол

Метод парабол является представителем группы методов, основанных на аппроксимации целевой функции некоторой другой функцией, точку минимума которой можно найти аналитически. Эта точка и принимается за очередное приближение искомого минимума целевой функции.

Пусть:

— $f(x)$ унимодальна на отрезке $[a,\ b]$;

— f достигает минимума во внутренней точке отрезка $[a,\ b]$.

Выберем точки $x_1,\ x_2,\ x_3 \in [a,\ b]$ так, чтобы

$$(*)\ \begin{cases} x_1 < x_2 < x_3, \\ f(x_1) \geqslant f(x_1) \leqslant f(x_1),\ \text{причем по крайней мере одно неравенство} \\ \text{должно быть строгим.} \end{cases} \tag{1.2}$$

Тогда в силу унимодальности функции f $x^* \in [x_1,\ x_3]$.

Аппроксимируем целевую функцию $f(x)$ параболой, проходящей через точки $(x_1,\ f_1)$, $(x_2,\ f_2)$, $(x_3,\ f_3)$, где $f_i = f(x_i)$, $i = 1, 2, 3$. Тогда в силу выбора точек $x_1,\ x_2,\ x_3$ ветви этой параболы будут направлены вверх, а точка $\overline{x}$ ее минимума будет принадлежать отрезку $[x_1,\ x_3]$. За очередное приближение точки x^* принимается точка $\overline{x}$.

Пусть $q(x) = a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2)$ — уравнение искомой параболы. Тогда можно показать, что

$$a_0 = f_1 \text{ (не будет использоваться)},$$

$$(**)\ \begin{cases} a_1 = \frac{f_2 - f_1}{x_2 - x_1}, \\ a_2 = \frac{1}{x_3 - x_2}\left[\frac{f_3 - f_1}{x_3 - x_1} - \frac{f_2 - f_1}{x_2 - x_1}\right], \\ \overline{x}\ = \frac{1}{2}\left[x_1 + x_2 - \frac{a_1}{a_2}\right]. \end{cases} \tag{1.3}$$

<u>Замечание</u>

1. О выборе точек $x_1,\ x_2,\ x_3$.

   (a) На первой итерации обычно достаточно несколько пробных точек. Можно выполнять итерации метода золотого сечения до тех пор, пока для двух пробных точек этого метода и одной из граничных точек очердного отрезка не будут выполнены неравенства $(*)$.

   (b) При второй и последующих итерациях на отрезке $[x_1,\ x_3]$ рассматриваются 2 пробные точки x_2 и $\overline{x}$, для которых используется метод исключения отрезков. В новом отрезке $[x_1',\ x_3']$ в качестве x_2' выбирается та точка из x_2 и $\overline{x}$, которая оказывается внутри.

2. На каждой итерации метода парабол, кроме первой, вычисляется 1 значение целевой функции: $\overline{f}$.

Схема рассматриваемого метода представлена на рисунке 1.1.

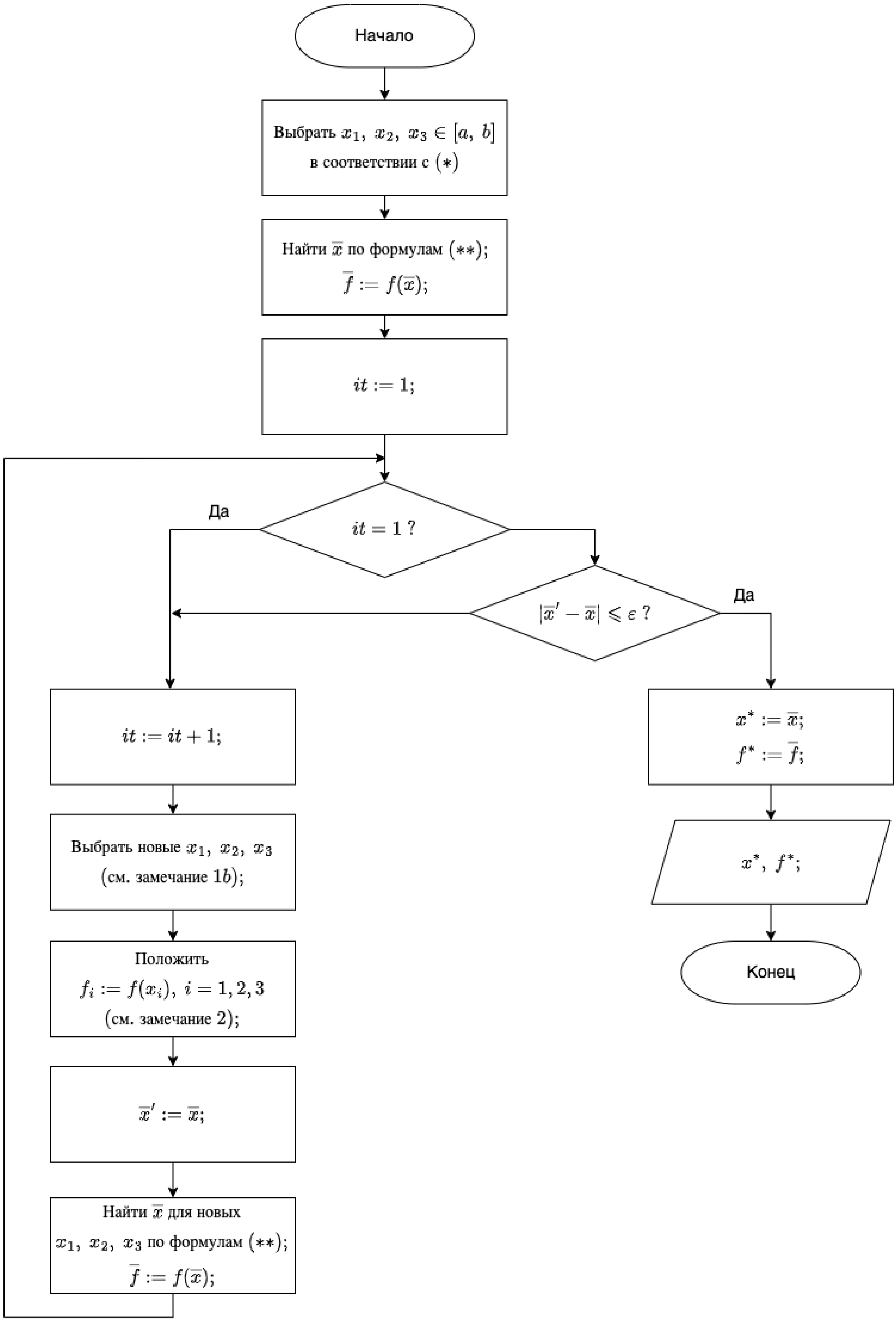

Рисунок 1.1 – Схема алгорима метода поразрядного поиска

2 Практическая часть

Таблица 2.1 – Результаты расчетов для задачи из индивидуального варианта

№ п/п	ε	N	x^*	$f(x^*)$
1	0.01	4	0.4789477465	-1.4738494147
2	0.0001	8	0.4824113669	-1.4738932842
3	0.000001	10	0.4824179876	-1.4738932844

В листинге 2.1 представлен код программы.

Листинг 2.1 — Код программы

```
function lab_03()
    clc();

    a = 0;
    b = 1;
    eps = 1e-6;

    debugFlag = true;
    delay = 0.5;

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

    fprintf('Выбор изначальных точек x1, x2, x3 с помощью метода золотого
↪   сечения:\n\n');

    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

```

Продолжение листинга 2.1

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

                x2 = x1;
                f2 = f1;

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

                if f1 >= f2
                    x3 = b;
                    f3 = fb;
                    break;
                end
            else
                a = x1;
                fa = f1;
                l = b - a;

                x1 = x2;
                f1 = f2;

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

                if f1 <= f2
                    x3 = x2;
                    f3 = f2;

                    x2 = x1;
                    f2 = f1;

                    x1 = a;
                    f1 = fa;
                    break;
                end
            end
            N = N + 1;
        else
            xStar = (a + b) / 2;
            fStar= f(xStar);
            N = N + 1;
```

Продолжение листинга 2.1

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

    N = N + 1;
    if debugFlag
        fprintf('N = %2d:   a = %.10f;   b = %.10f;\n\n', N, a, b);
        line([a b], [f(a) f(b)]);
        pause(delay);
        scatter(x1, f1, 'b', 'filled');
        scatter(x2, f2, 'b', 'filled');
        scatter(x3, f3, 'b', 'filled');
        fprintf('x1 = %.10f\nx2 = %.10f\nx3 = %.10f\n\n', x1, x2, x3);
        line([x1 x3], [f1 f3], 'color', 'b');
        pause(delay);
    end

    fprintf('Метод парабол:\n\n');

    a1 = (f2 - f1) ./ (x2 - x1);
    a2 = ((f3 - f1) ./ (x3 - x1) - (f2 - f1) ./ (x2 - x1)) ./ (x3 - x2);
    x_line = 1 / 2 .* (x1 + x2 - a1 ./ a2);
    f_line = f(x_line);
    it = 1;

    while true
        if it == 1 || abs(prev_x_line - x_line) > eps
            it = it + 1;

            if f_line > f2
                temp = f_line;
                f_line = f2;
                f2 = temp;

                temp = x_line;
                x_line = x2;
                x2 = temp;
            end

            if x_line > x2
                x1 = x2;
                f1 = f2;

```

Продолжение листинга 2.1

```
                x2 = x_line;
                f2 = f_line;
            else
                x3 = x2;
                f3 = f2;

                x2 = x_line;
                f2 = f_line;
            end

            prev_x_line = x_line;
            a1 = (f2 - f1) ./ (x2 - x1);
            a2 = ((f3 - f1) ./ (x3 - x1) - (f2 - f1) ./ (x2 - x1)) ./ (x3 - x2);
            x_line = 1 / 2 .* (x1 + x2 - a1 ./ a2);
            f_line = f(x_line);

            N = N + 1;
            if debugFlag
                fprintf('Для N = %2d:\n', N);
                fprintf('x1 = %.10f;   f1 = %.10f;\n', x1, f1);
                fprintf('x2 = %.10f;   f2 = %.10f;\n', x2, f2);
                fprintf('x3 = %.10f;   f3 = %.10f;\n', x3, f3);
                fprintf('Текущее приближение: x = %.10f, f(x) = %.10f\n\n', x_line,
    f_line);
                line([x1 x3], [f1 f3], 'color', 'b');
                pause(delay);
            end
        else
            xStar = x_line;
            fStar = f_line;
            break;
        end
    end

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