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, искусственный интеллект и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЁТ

по лабораторной работе № 3

по курсу «Методы вычислений»

на тему: «Метод парабол»

Вариант № 6

Студент	ИУ7-23М (Группа)	(Подпись, дата)	Керимов А. Ш. (Фамилия И. О.)
Преподаватель		(Подпись, дата)	Власов П. А. (Фамилия И. О.)

2022 г.

Постановка задачи

Решить одномерную задачу оптимизации вида

$$\begin{cases} f(x) \to \min, \\ x \in [a, b], \end{cases} \tag{1}$$

методом парабол с заданной точностью ε.

Входные данные

Заданная функция:

$$f(x) = \operatorname{ch}\left(\frac{3x^3 + 2x^2 - 4x + 5}{3}\right) + \operatorname{th}\left(\frac{x^3 - 3\sqrt{2}x - 2}{2x + \sqrt{2}}\right) - 2{,}5. \tag{2}$$

Поиск точки минимума производится на отрезке $[0, 1]$. При построении таблицы результатов в качестве точности ε были взяты следующие значения: $\{10^{-2}, 10^{-4}, 10^{-6}\}$.

Метод парабол

Схема метода парабол представлена на рисунке 1.

Условия выбора точек $x_1, x_2, x_3 \in [a, b)$:

1. $x_1 < x_2 < x_3$,
2. $f(x_1) \geqslant f(x_2) \leqslant f(x_3)$ принимает по крайней мере одно неравенство строгое.

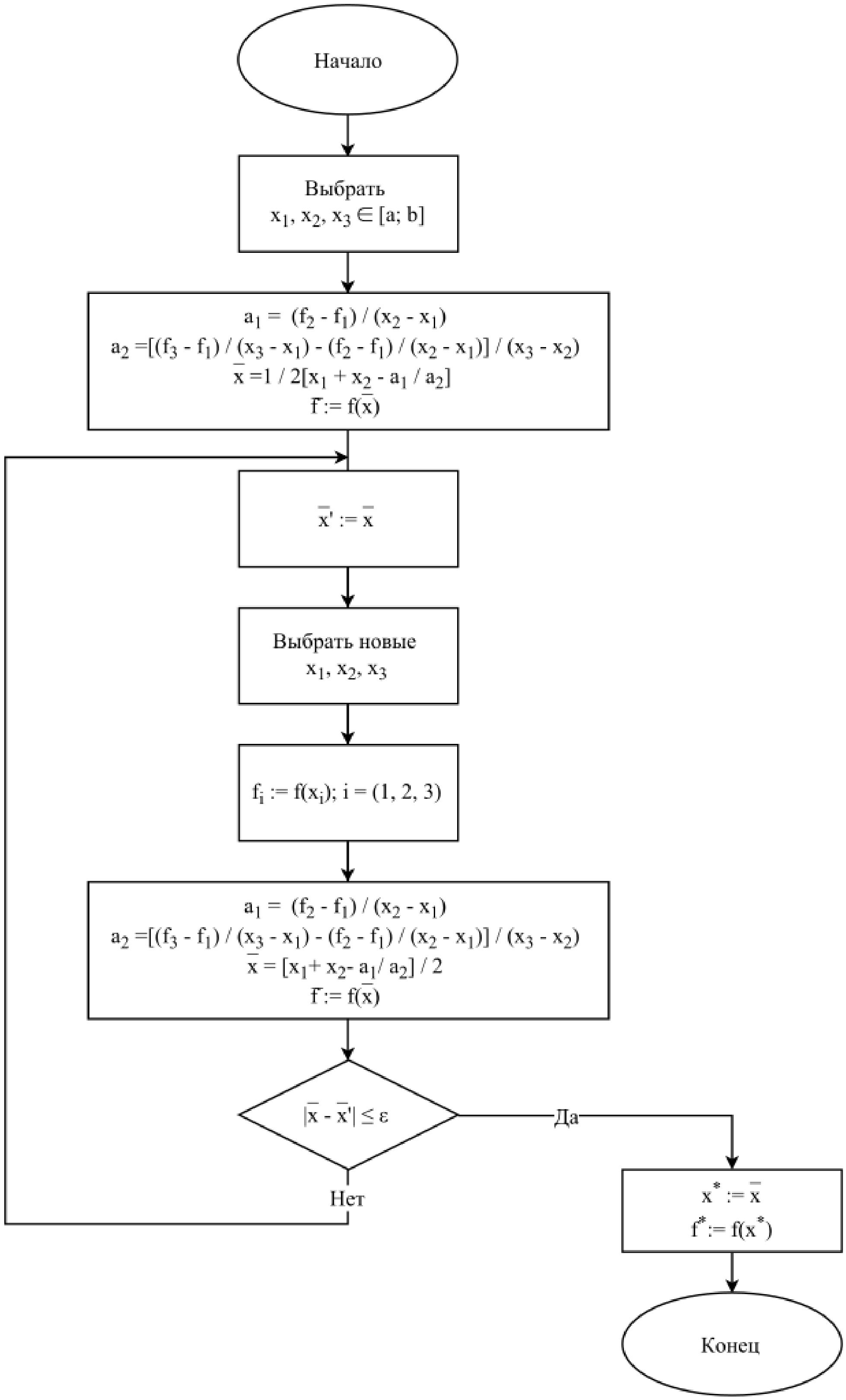

Рисунок 1 — Схема метода парабол

Замечания:

1. В качестве критерия окончания вычислений используется условие $|\overline{x} - \overline{x}'| < \varepsilon$, означающее близость друг к другу двух последовательных приближений точки x^* . Вообще говоря, выполнение этого условия не гарантирует близость этих точек к x^*. Однако на практике такое условие удовлетворительно работает. Дополнительно точность текущего приближения можно оценивать (если получится) с использованием длины отрезка $[x_1, x_3]$.

2. О выборе точек x_1, x_2, x_3

   (a) Можно выполнить несколько итераций метода золотого сечения до тех пор, пока пробные точки этого метода и одна из граничных точек текущего отрезка не будут удовлетворять условиям 1 и 2.

   (b) На второй и последующих итерациях на отрезке $[x_1, x_3]$ рассматриваются две 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 оказалась внутри.

3. На каждой итерации метода парабол, кроме первой, вычисляется только одно значение целевой функции $\overline{f}$.

Результаты вычислений

Таблица 1 — Результаты вычислений

№ п/п	ε	N	x^*	$f(x^*)$
1	10^{-2}	9	0,4773394983	−1,4737994041
2	10^{-4}	13	0,4824088798	−1,4738932840
3	10^{-6}	15	0,4824178751	−1,4738932844

Текст программы

Листинг 1 — lab03.m

```
function lab03
    debug = true;

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

    fplot(@(x) func(x), [a, b], 'b');
    hold on;

    run = 'p'; % p - parabola
               % g - golden ratio
               % a - all
    p1 = 0;
    p2 = 0;
    m1 = 'Метод парабол';
    m2 = 'Метод золотого сечения';
    global N;

    if (run == 'p' || run == 'a')
        N = 0;
        disp('Метод парабол');
        [x, f] = successiveParabolicInterpolation(a, b, eps, debug);
        fprintf('Минимум функции: (x=%12.10f, f=%12.10f)\n', x, f);
        fprintf('N = %d\n', N);
        p1 = plot(x, f, 'rx', 'MarkerSize', 15);
        if (run ~= a)
            legend((p1), m1, 'Location', 'northwest');
        end
    end
    if (run == 'g' || run == 'a')
        N = 0;
        disp('Метод золотого сечения');
        [x, f, ~, ~] = goldenSectionSearch(a, b, eps, debug);
        fprintf('Минимум функции: (x=%12.10f, f=%12.10f)\n', x, f);
        fprintf('N = %d\n', N);
        p2 = plot(x, f, 'bx', 'MarkerSize', 15);
        if (run ~= 'a')
            legend((p2), m2, 'Location', 'northwest');
        end
    end
    if (run == 'a')
        legend([p1 p2], {m1 m2}, 'Location', 'northwest');
    end

    hold off;
end

function y = func(x)
    global N;
    N = N + 1;

    x3 = power(x, 3);
    x2 = power(x, 2);
    sqrt2 = sqrt(2);
```

```

    ch = cosh((3 * x3 + 2 * x2 - 4 * x + 5) / 3);
    th = tanh((x3 - 3 * sqrt2 * x - 2) / (2 * x + sqrt2));

    y = ch + th - 2.5;
end

function [x, f] = successiveParabolicInterpolation(a, b, eps, debug)
    [xl, xr] = goldenSectionBoundaries(a, b);
    xm = (xl + xr) / 2;
    fl = func(xl);
    fm = func(xm);
    fr = func(xr);

    [x, f] = parabola(xl, fl, xm, fm, xr, fr);

    run = true;
    iteration = 1;

    while (run)
        if (debug)
            fprintf('Итерация %d: [x1=%12.10f, x3=%12.10f]\n', iteration, xl, xr);
            iteration = iteration + 1;
        end

        if (x < xm)
            if (f >= fm)
                xl = x;
                fl  = f;
            else
                xr = xm;
                fr = fm;
                xm = x;
                fm = f;
            end
        else
            if (f >= fm)
                xr = x;
                fr = f;
            else
                xl = xm;
                fl = fm;
                xm = x;
                fm = f;
            end
        end

        x0 = x;
        [x, f] = parabola(xl, fl, xm, fm, xr, fr);
        run = abs(x - x0) > eps;
    end

    if (debug)
        fprintf('Итерация %d: [x1=%10.8f, x3=%10.8f]\n', iteration, xl, xr);
    end
end

```

```
function [x, f] = parabola(xl, fl, xm, fm, xr, fr)
    a1 = (fm - fl) / (xm - xl);
    a2 = ((fr - fl) / (xr - xl) - a1) / (xr - xm);
    x = (xl + xm - a1 / a2) / 2;
    f = func(x);
end

function [xl, xr] = goldenSectionBoundaries(a, b)
    [~, ~, xl, xr] = goldenSectionSearch(a, b, 0.49, false);
end

function [x, f, x1, x2] = goldenSectionSearch(a, b, eps, debug)
    tau = (sqrt(5) - 1) / 2;
    delta = b - a;

    xl = b - tau * delta;
    xr = a + tau * delta;
    fl = func(xl);
    fr = func(xr);

    iteration = 1;
    while (delta > 2 * eps)
        if (debug)
            fprintf('Итерация %d: [a=%12.10f, b=%12.10f], (xl=%12.10f, xr=%12.10f)\n',
                iteration, a, b, xl, xr);
            iteration = iteration + 1;
        end

        if (fl > fr)
            a = xl;
            delta = b - a;
            xl = xr;
            fl = fr;
            xr = a + tau * delta;
            fr = func(xr);
        else
            b = xr;
            delta = b - a;
            xr = xl;
            fr = fl;
            xl = b - tau * delta;
            fl = func(xl);
        end
    end

    if (debug)
        fprintf('Итерация %d: [a=%12.10f, b=%12.10f], (xl=%12.10f, xr=%12.10f)\n', iteration,
            a, b, xl, xr);
    end

    x = (a + b) / 2;
    f = func(x);

    x1 = a;
    x2 = b;
end
```