



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  
(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 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:

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

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 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

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

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  $\bar{x}$  ее минимума будет принадлежать отрезку  $[x_1, x_3]$ . За очередное приближение точки  $x^*$  принимается точка  $\bar{x}$ .

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

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

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

#### Замечание

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 очередного отрезка не будут выполнены неравенства (\*).

(b) При второй и последующих итерациях на отрезке  $[x_1, x_3]$  рассматриваются 2 пробные точки  $x_2$  и  $\bar{x}$ , для которых используется метод исключения отрезков. В новом отрезке  $[x'_1, x'_3]$  в качестве  $x'_2$  выбирается та точка из  $x_2$  и  $\bar{x}$ , которая оказывается внутри.

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

Схема рассматриваемого метода представлена на рисунке 1.1.

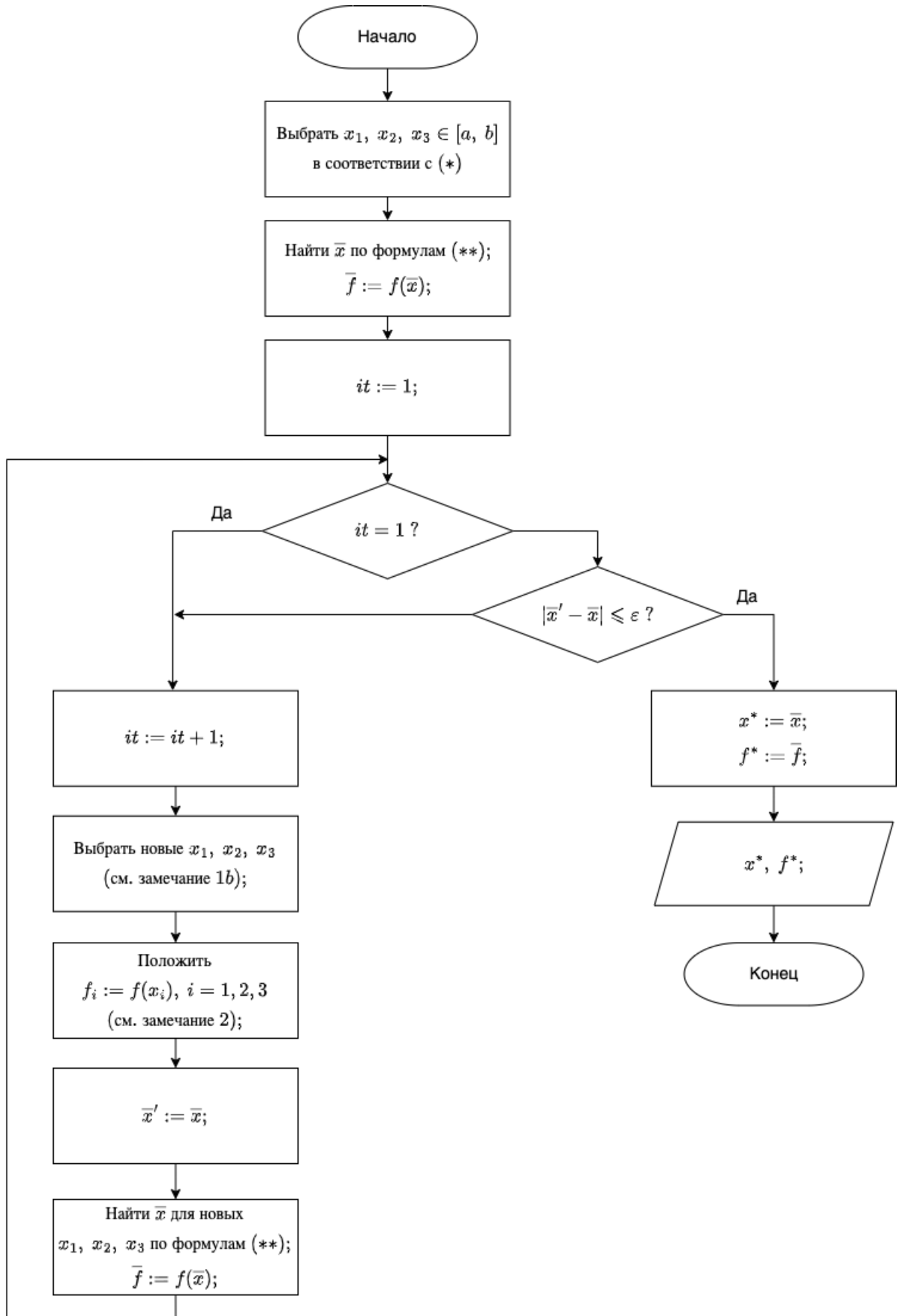


Рисунок 1.1 – Схема алгоритма метода поразрядного поиска

## 2 Практическая часть

Таблица 2.1 – Результаты расчетов для задачи из индивидуального варианта

№ п/п	$\varepsilon$	$N$	$x^*$	$f(x^*)$
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

```
1  function lab_03()
2      clc();
3
4      a = 0;
5      b = 1;
6      eps = 1e-6;
7
8      debugFlag = true;
9      delay = 0.5;
10
11     fplot(@f, [a, b]);
12     hold on;
13
14     fprintf('Выбор изначальных точек x1, x2, x3 с помощью метода золотого
↪ сечения:\n\n');
15
16     tau = (sqrt(5) - 1) / 2;
17     l = b - a;
18
19     x1 = b - tau * l;
20     x2 = a + tau * l;
21     f1 = f(x1);
22     f2 = f(x2);
23
24     N = 2;
25     while true
26         if debugFlag
27             fprintf('N = %2d:   a = %.10f;   b = %.10f;\n', N, a, b);
28             line([a b], [f(a) f(b)]);
29             pause(delay);
30         end
31     end
```

## Продолжение листинга 2.1

```
32         if l > 2 * eps
33             if f1 <= f2
34                 b = x2;
35                 fb = f2;
36                 l = b - a;
37
38                 x2 = x1;
39                 f2 = f1;
40
41                 x1 = b - tau * l;
42                 f1 = f(x1);
43
44             if f1 >= f2
45                 x3 = b;
46                 f3 = fb;
47                 break;
48             end
49         else
50             a = x1;
51             fa = f1;
52             l = b - a;
53
54             x1 = x2;
55             f1 = f2;
56
57             x2 = a + tau * l;
58             f2 = f(x2);
59
60             if f1 <= f2
61                 x3 = x2;
62                 f3 = f2;
63
64                 x2 = x1;
65                 f2 = f1;
66
67                 x1 = a;
68                 f1 = fa;
69                 break;
70             end
71         end
72         N = N + 1;
73     else
74         xStar = (a + b) / 2;
75         fStar = f(xStar);
76         N = N + 1;
```

## Продолжение листинга 2.1

```

77         scatter(xStar, fStar, 'r', 'filled');
78         fprintf('\nОтвет:   x* = %.10f;   f(x*) = %.10f.\n\n', xStar, fStar);
79         return;
80     end
81 end
82
83 N = N + 1;
84 if debugFlag
85     fprintf('N = %2d:   a = %.10f;   b = %.10f;\n\n', N, a, b);
86     line([a b], [f(a) f(b)]);
87     pause(delay);
88     scatter(x1, f1, 'b', 'filled');
89     scatter(x2, f2, 'b', 'filled');
90     scatter(x3, f3, 'b', 'filled');
91     fprintf('x1 = %.10f\nx2 = %.10f\nx3 = %.10f\n\n', x1, x2, x3);
92     line([x1 x3], [f1 f3], 'color', 'b');
93     pause(delay);
94 end
95
96 fprintf('Метод парабол:\n\n');
97
98 a1 = (f2 - f1) ./ (x2 - x1);
99 a2 = ((f3 - f1) ./ (x3 - x1) - (f2 - f1) ./ (x2 - x1)) ./ (x3 - x2);
100 x_line = 1 / 2 .* (x1 + x2 - a1 ./ a2);
101 f_line = f(x_line);
102 it = 1;
103
104 while true
105     if it == 1 || abs(prev_x_line - x_line) > eps
106         it = it + 1;
107
108         if f_line > f2
109             temp = f_line;
110             f_line = f2;
111             f2 = temp;
112
113             temp = x_line;
114             x_line = x2;
115             x2 = temp;
116         end
117
118         if x_line > x2
119             x1 = x2;
120             f1 = f2;
121

```

## Продолжение листинга 2.1

```

122         x2 = x_line;
123         f2 = f_line;
124     else
125         x3 = x2;
126         f3 = f2;
127
128         x2 = x_line;
129         f2 = f_line;
130     end
131
132     prev_x_line = x_line;
133     a1 = (f2 - f1) ./ (x2 - x1);
134     a2 = ((f3 - f1) ./ (x3 - x1) - (f2 - f1) ./ (x2 - x1)) ./ (x3 - x2);
135     x_line = 1 / 2 .* (x1 + x2 - a1 ./ a2);
136     f_line = f(x_line);
137
138     N = N + 1;
139     if debugFlag
140         fprintf('Для N = %2d:\n', N);
141         fprintf('x1 = %.10f;   f1 = %.10f;\n', x1, f1);
142         fprintf('x2 = %.10f;   f2 = %.10f;\n', x2, f2);
143         fprintf('x3 = %.10f;   f3 = %.10f;\n', x3, f3);
144         fprintf('Текущее приближение: x = %.10f, f(x) = %.10f\n\n', x_line,
↪ f_line);
145         line([x1 x3], [f1 f3], 'color', 'b');
146         pause(delay);
147     end
148     else
149         xStar = x_line;
150         fStar = f_line;
151         break;
152     end
153 end
154
155 if debugFlag
156     scatter(xStar, fStar, 'r', 'filled');
157     fprintf('Ответ:   x* = %.10f;   f(x*) = %.10f.\n\n', xStar, fStar);
158 end
159 end
160
161 function y = f(x)
162     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;
163 end

```