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 №3
по курсу «Методы вычислений»
на тему: «Метод парабол»
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 парабол для решения задачи одномерной минимизации.

Задание:

1. Реализовать метод парабол 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 отрезков $[x_{1i}, x_{3i}]$, содержащих точку искомого минимума (для последовательности отрезков следует предусмотреть возможность «отключения» вывода ее на экран).

1.1 Исходные данные варианта №7

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

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

1.2 Краткое описание метода парабол

Метод парабол является представителем группы методов, основанных на аппроксимации целевой функции некоторой более простой функцией (как правило полиномом), минимум которой можно легко найти. Точка минимума этой аппроксимируещей функции и принимается за очередное приближение точки минимума целевой функции.

Пусть

1. f унимодальна на $[a; b]$,

2. f минимума во внутренней точке отрезка $[a; b]$.

Выберем три точки $x_1, x_2, x_3 \in [a; b]$, так чтобы (*):

1. $x_1 < x_2 < x_3$,
2. $f(x_1) \geq f(x_2) \leq f(x_3)$ — хотя бы одно из неравенств строгое.

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

Аппроксимируем целевую функцию параболой, проходящей через точки (x_1, f_1), (x_2, f_2), (x_3, f_3), где $f_i = f(x_i)$.

В силу условий (*) ветви параболы направленны вверх. Это значит, что точка $\overline{x}$ минимума этой параболы также принадлежит отрезку [x1, x3]. Точка $\overline{x}$ принимается за очередное приближение точки x^*.

Пусть $q(x) = a_0 + a_1 \cdot (x - x_1) + a_2 \cdot (x - x_1) \cdot (x - x_2)$ — уравнение параболы.

Можно показать, что условия $q(x_i) = f_i$, приводят к (**):

$$a_0 = f_1,$$

$$a_1 = \frac{f_2 - f_1}{x_2 - x_1},$$

$$a_2 = \frac{1}{x_3 - x_2} \cdot \left(\frac{f_3 - f_1}{x_3 - x_1} - \frac{f_2 - f_1}{x_2 - x_1} \right),$$

$$\overline{x} = \frac{1}{2} \cdot \left(x_1 + x_2 - \frac{a_1}{a_2} \right).$$

О выборе точек.

1. На первой итерации для выбора точек x_1, x_2, x_3 обычно достаточно использование нескольких пробных точек. Если это не получается за разумное время, 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 (**).

2. 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.

Вычисления продолжаются до тех пор, пока не выполнится условие $|\overline{x} - \overline{x}'|$, где $\overline{x}'$ — приближение x^* с предыдущей итерации.

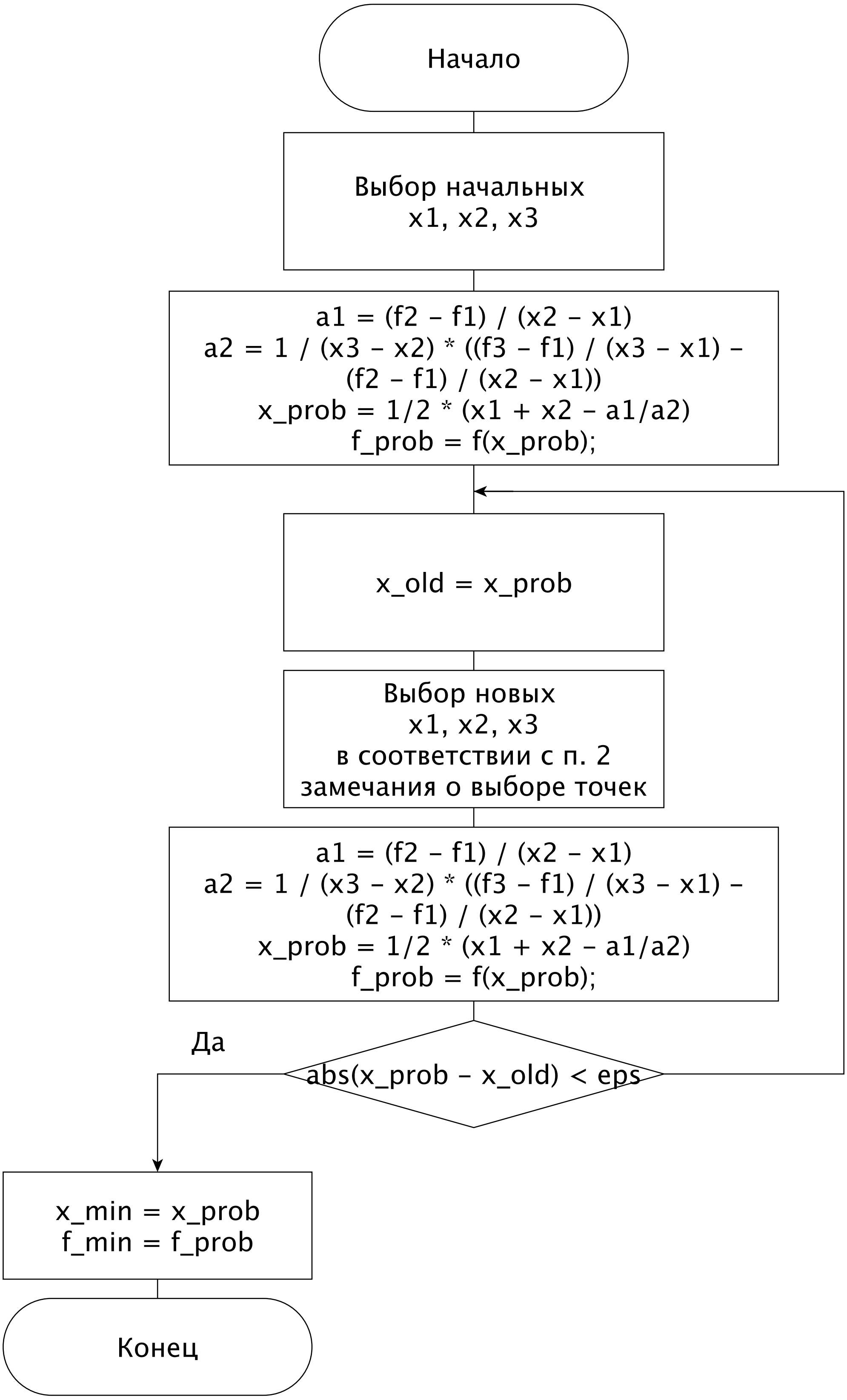

Рисунок 1.1 – Схема алгоритма метода парабол

2 Практический раздел

Листинг 2.1 – Исходный код программы

```
# Лабораторная работа 3. Вариант 7.

function main()
  clc;

  debug = false;

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

  [x_min, f_min, n, xs, fs, as, bs] = find_min(debug, a, b, eps);
  draw_plot(debug, a, b, eps, x_min, f_min, xs, fs, as, bs);
  fprintf('\n\033[36mТочка минимума (x*, f(x*)) = (%.10f, %.10f), количество вычислений функции: %d.\033[0m\n', x_min, f_min, n);
end

function [x1, x2, x3, f1, f2, f3, n] = find_x123(debug, a, b)
  tau = (sqrt(5) - 1) / 2;

  l = b - a;

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

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

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

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

  i = 2;
```

```

fa = f(a);
fb = f(b);

while true
  if f1 <= f2
    b = x2;
    fb = f2;
    l = b - a;

    x2 = x1;
    f2 = f1;

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

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

    x1 = x2;
    f1 = f2;

    x2 = a + tau*l;
    f2 = f(x2);
    i = i + 1;

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

  if fa > f1 && f1 < f2
    x3 = x2;
```

```
      f3 = f2;

      x2 = x1;
      f2 = f1;

      x1 = a;
      f1 = fa;

      n = i;
      return;
    elseif f1 > f2 && f2 < fb
      x3 = b;
      f3 = fb;

      n = i;
      return;
    endif
  endwhile
end

function [x_min, f_min, n, xs, fs, as, bs] = find_min(debug, a, b, eps)
  [x1, x2, x3, f1, f2, f3, n] = find_x123(debug, a, b);

  xs = [];
  fs = [];
  as = [];
  bs = [];

  as(end + 1) = x1;
  bs(end + 1) = x3;

  a1 = (f2 - f1) / (x2 - x1);
  a2 = 1 / (x3 - x2) * ((f3 - f1) / (x3 - x1) - (f2 - f1) / (x2 - x1));
  x_prob = 1/2 * (x1 + x2 - a1/a2);
  f_prob = f(x_prob);
  n = n + 1;

  if debug
    fprintf('Итерация %d: (x1, f1) = (%.10f, %.10f), (x2, f2) =
```

```
    (%.10f, %.10f), (x3, f3) = (%.10f, %.10f), (x_prob,
    f_prob) = (%.10f, %.10f).\n', n-1, x1, f1, x2, f2, x3,
    f3, x_prob, f_prob);
endif

xs(end + 1) = x_prob;
fs(end + 1) = f_prob;

while true
  x_old = x_prob;

  if x2 < x_prob
    if f2 < f_prob
      x3 = x_prob;
      f3 = f_prob;
    else
      x1 = x2;
      f1 = f2;
      x2 = x_prob;
      f2 = f_prob;
    endif
  else
    if f_prob < f2
      x3 = x2;
      f3 = f2;
      x2 = x_prob;
      f2 = f_prob;
    else
      x1 = x_prob;
      f1 = f_prob;
    endif
  endif

  as(end + 1) = x1;
  bs(end + 1) = x3;

  a1 = (f2 - f1) / (x2 - x1);
  a2 = 1 / (x3 - x2) * ((f3 - f1) / (x3 - x1) - (f2 - f1) /
     (x2 - x1));
  x_prob = 1/2 * (x1 + x2 - a1/a2);
  f_prob = f(x_prob);
```

```
    n = n + 1;
    if debug
      fprintf('Итерация %d: (x1, f1) = (%.10f, %.10f), (x2, f2) = (%.10f, %.10f), (x3, f3) = (%.10f, %.10f), (x_prob, f_prob) = (%.10f, %.10f).\n', n-1, x1, f1, x2, f2, x3, f3, x_prob, f_prob);
    endif

    xs(end + 1) = x_prob;
    fs(end + 1) = f_prob;

    if abs(x_prob - x_old) < eps
      x_min = x_prob;
      f_min = f_prob;
      return;
    endif
  endwhile
end

function draw_plot(debug, a, b, step, x_min, f_min, xs, fs, as, bs)
  sleep_seconds = 0.1;

  fprintf('as %d, xs %d\n', length(as), length(xs));

  x = a:step:b;
  y = zeros(size(x));
  for i = 1:length(x)
      y(i) = f(x(i));
  end
  plot(x,y);
  hold on;

  if debug
    scatter(xs(1), fs(1), 8, 'r', 'filled');
    line([as(1), bs(1)], [f(as(1)), f(bs(1))], 'DisplayName', sprintf('шаг %d', 1), 'Color', 'r');
    pause(sleep_seconds);

    for i = 2:length(xs)
        scatter(xs(i-1), fs(i-1), 8, 'b', 'filled');
```

```
      scatter(xs(i), fs(i), 8, 'r', 'filled');
      pause(sleep_seconds);

      if i < length(as)
        line([as(i), bs(i)], [f(as(i)), f(bs(i))],
           'DisplayName', sprintf('шаг %d', i), 'Color', 'r');
      endif

      pause(sleep_seconds);
    end
  endif

  scatter(x_min, f_min, 10, 'g', 'filled');
  text(x_min, f_min, sprintf('\n\n\n\n(%.10f, %.10f)', 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}	5	1.3213785811	-0.4609645405
2	10^{-4}	9	1.3211624486	-0.4609645959
3	10^{-6}	12	1.3211613308	-0.4609645959