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 №4
по курсу «Методы вычислений»
на тему: «Метод Ньютона»
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 модифицированный метод Ньютона с конечно-разностной аппроксимацией производных 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 и последовательности точек $(x_i, f(x_i))$, аппроксимирующих точку искомого минимума (для последовательности точек следует предусмотреть возможность «отключения» вывода ее на экран);

4. провести решение задачи с использованием стандартной функции `fminbnd` пакета MatLAB.

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 Метод Ньютона

Пусть $f(x)$ — дважды дифференцируема на отрезке $[a; b]$ и выпукла. Следовательно, она унимодальна, и условие $f'(x)$ — необходимо и достаточно для поиска ее минимума.

Рассмотрим решение уравнения $f'(x) = 0$ и его решение методом Ньютона.

Основная идея метода Ньютона: за очередное приближение корня уравнения принимается точка пересечения с осью x касательной к графику функции в точке, отвечающей текущему приближению.

Уравнение касательной к графику функции $f'(x)$ в точке x_0 имеет вид $y = f'(x_0) + f''(x_0) * (x - x_0)$. Тогда расчетное соотношение можно записать как $x_k = x_{k-1} - \frac{f'(x_{k-1})}{f''(x_{k-1})}$, где x_k — текущее приближение, x_{k-1} — предыдущее.

Вычисления продолжаются до тех пор, пока не выполнится одно из условий: $|\overline{x} - \overline{x}'| <= \epsilon$, где $\overline{x}'$ — приближение x^* с предыдущей итерации, или $|f'(x_i)| <= \epsilon$.

1.3 Модифицированный метод Ньютона

Если вычисление производных трудоемко, то используется модифицированный метод Ньютона. Тогда в качестве очередного приближения x^* используется точка пересечения с осью x прямой, параллельной касательной к графику функции в точке x_0 — первого приближения.

Тогда расчетное соотношение можно записать как $x_k = x_{k-1} - \frac{f'(x_{k-1})}{f''(x_0)}$, где x_k — текущее приближение, x_{k-1} — предыдущее, x_0 — первое приближение.

Вместо вычисления производных используются конечно-разностные аппроксимации:

$$f'(x_i) = \frac{f(x_i + h) - f(x_i - h)}{2h},$$

$$f''(x_i) = \frac{f(x_i + h) - 2f(x_i) + f(x_i - h)}{h^2},$$

где h — малая величина.

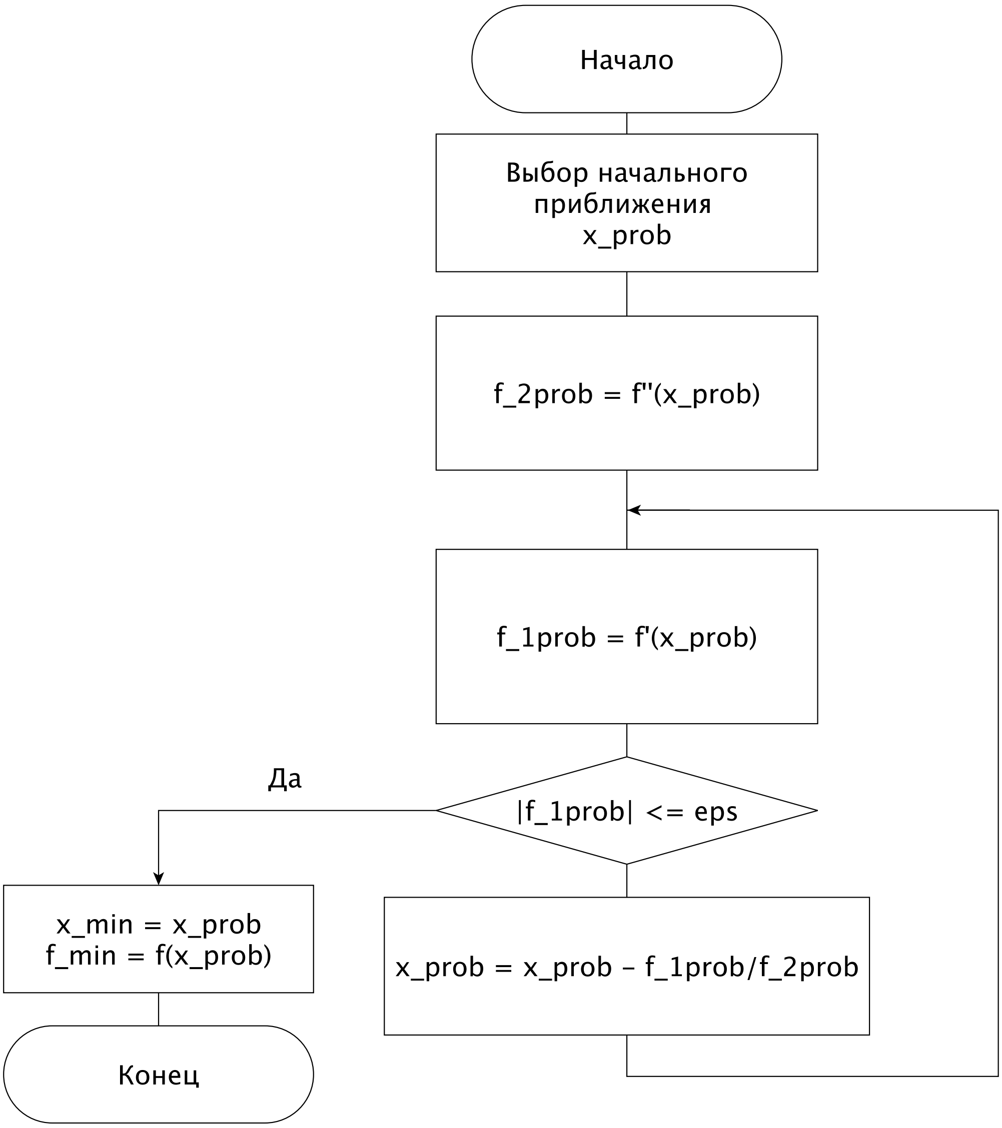

Рисунок 1.1 – Схема алгоритма модифицированного метода Ньютона

2 Практический раздел

Листинг 2.1 – Исходный код программы

```
# Лабораторная работа 4. 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);
  fprintf('\n\033[36mТочка минимума (x*, f(x*)) = (%.10f,
     %.10f), количество вычислений функции: %d.\033[0m\n',
     x_min, f_min, n);

  options = optimset('TolX', eps);
  if debug
    options = optimset(options, 'Display', 'iter');
  end
  [x_min_default, f_min_default] = fminbnd(@f, a, b, options);
  fprintf('\n\033[36mТочка минимума методом fminbnd (x*, f(x*))
     = (%.10f, %.10f).\033[0m\n', x_min_default, f_min_default);

  draw_plot(debug, a, b, eps, x_min, f_min, xs, fs);
end

function [x_min, f_min, n, xs, fs] = find_min(debug, a, b, eps)
  [x_prob, f_prob, n] = find_x0(debug, a, b);

  #x_prob = (a + b) / 2;
  #f_prob = f(x_prob);
  #n = 1;

  xs = [];
  fs = [];

  xs(end + 1) = x_prob;
```

```
  fs(end + 1) = f_prob;

  delta = 1e-3;

  i = n - 1;
  if debug
    fprintf('Итерация %d: (x, f) = (%.10f, %.10f).\n', i, x_prob, f_prob);
  endif
  i = i + 1;

  f2 = (f(x_prob - delta) - 2 * f_prob + f(x_prob + delta)) / (delta ^ 2);
  n = n + 2;

  while true
    f1 = (f(x_prob + delta) - f(x_prob - delta)) / (2 * delta)
    n = n + 2;

    if abs(f1) < eps
      x_min = x_prob;
      f_min = f(x_prob);
      n = n + 1;
      return;
    endif

    x_prob = x_prob - f1/f2;
    xs(end + 1) = x_prob;
    fs(end + 1) = f(x_prob);
    if debug
      fprintf('Итерация %d: (x, f) = (%.10f, %.10f).\n', i, x_prob, f(x_prob));
    endif
    i = i+1;
  endwhile
end

function draw_plot(debug, 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;
  if debug
    scatter(xs(1), fs(1), 8, 'r', 'filled');
    for i = 2:length(xs)
        scatter(xs(i-1), fs(i-1), 8, 'b', 'filled');
        scatter(xs(i), fs(i), 8, 'r', 'filled');
        pause(0.5);
    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

function [x0, f0, n] = find_x0(debug, a, b)
  iterations = 4;

  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;

for j = 1:iterations
  if f1 <= f2
    b = x2;
    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;
    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
endfor

n = i + 1;
x0 = (a + b) / 2;
```

```
  f0 = f(x0);
end
```

Таблица 2.1 – Результаты расчетов по индивидуальному варианту

№	ϵ	N	x^*	$f(x^*)$
1	10^{-2}	14	1.3209353010	-0.4609645359
2	10^{-4}	16	1.3211535883	-0.4609645959
3	10^{-6}	18	1.3211615186	-0.4609645959

Таблица 2.2 – Сводная таблица по лабораторным работам для $\epsilon = 10^{-6}$

№ п/п	Метод	N	x^*	$f(x^*)$
1	Поразрядного поиска	55	1.3211612701	-0.4609645959
2	Золотого сечения	31	1.3211617177	-0.4609645959
3	Парабол	12	1.3211614113	-0.4609645959
4	Ньютона модифицированный	18	1.3211615186	-0.4609645959
5	Функция fminbnd	9	1.3211613111	-0.4609645959