



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 №4

по курсу «Методы вычислений»

на тему: «Метод Ньютона»

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 модифицированный метод Ньютона с конечно-разностной аппроксимацией производных 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);
4. провести решение задачи с использованием стандартной функции `fminbnd` пакета *MatLAB*.

Таблица 1.1 – Данные индивидуального варианта

№ вар.	Целевая функция $f(x)$	[a, b]
6	$\text{ch} \left(\frac{3x^3 + 2x^2 - 4x + 5}{3} \right) + \text{th} \left(\frac{x^3 - 3\sqrt{2}x - 2}{2x + \sqrt{2}} \right) - 2.5$	[0, 1]

1.1 Краткое описание метода Ньютона

Для приближения корня x^* используется точка \bar{x} пересечения касательной к графику $g(x)$ в точке текущего приближения с осью Ox .

Пусть \bar{x} — текущее приближение точки x^* . Уравнение касательной к графику функции g в точке $(\bar{x}, g(\bar{x}))$:

$$y - \bar{y} = g'(\bar{x})(x - \bar{x}). \quad (1.2)$$

Пересечение с OX :

$$\bar{x} = \bar{x}' - \frac{g(\bar{x}')}{g'(\bar{x}')}, \quad (1.3)$$

где \bar{x}' — приближение x^* с предыдущего шага, \bar{x} — приближение x^* с текущего шага.

Условие окончания итераций:

$$|\bar{x} - \bar{x}'| < \varepsilon \text{ или } |g(\bar{x})| < \varepsilon. \quad (1.4)$$

Замечание

1. Метод Ньютона обладает высокой точностью и скоростью сходимости, если начальное приближение выбрано удачно. Если нет — метод может разойтись. Тогда стоит выполнить несколько итераций другого метода, например, золотого сечения.

2. Модифицированный метод Ньютона.

Расчетная схема метода:

$$\bar{x} = \bar{x}' - \frac{g(\bar{x}')}{g'(\bar{x}')}, \quad (1.5)$$

где x_0 — начальное приближение корня.

В качестве очередного приближения для корня x^* используется точка пересечения прямой, проходящей через x_n и параллельной касательной, с осью Ox в точке x_0 .

В данном методе меньше вычислений в рамках одной итерации, но самих итераций больше.

3. О вычислении производных.

Конечно-разностная аппроксимация производных:

$$f'(\bar{x}) \approx \frac{f(\bar{x} + \delta) - f(\bar{x} - \delta)}{2\delta}, \delta > 0; \quad (1.6)$$

$$f''(\bar{x}) \approx \frac{f(\bar{x} - \delta) - 2f(\bar{x}) + f(\bar{x} + \delta)}{\delta^2}, \delta > 0. \quad (1.7)$$

Схема рассматриваемого метода представлена на рисунке 1.1.

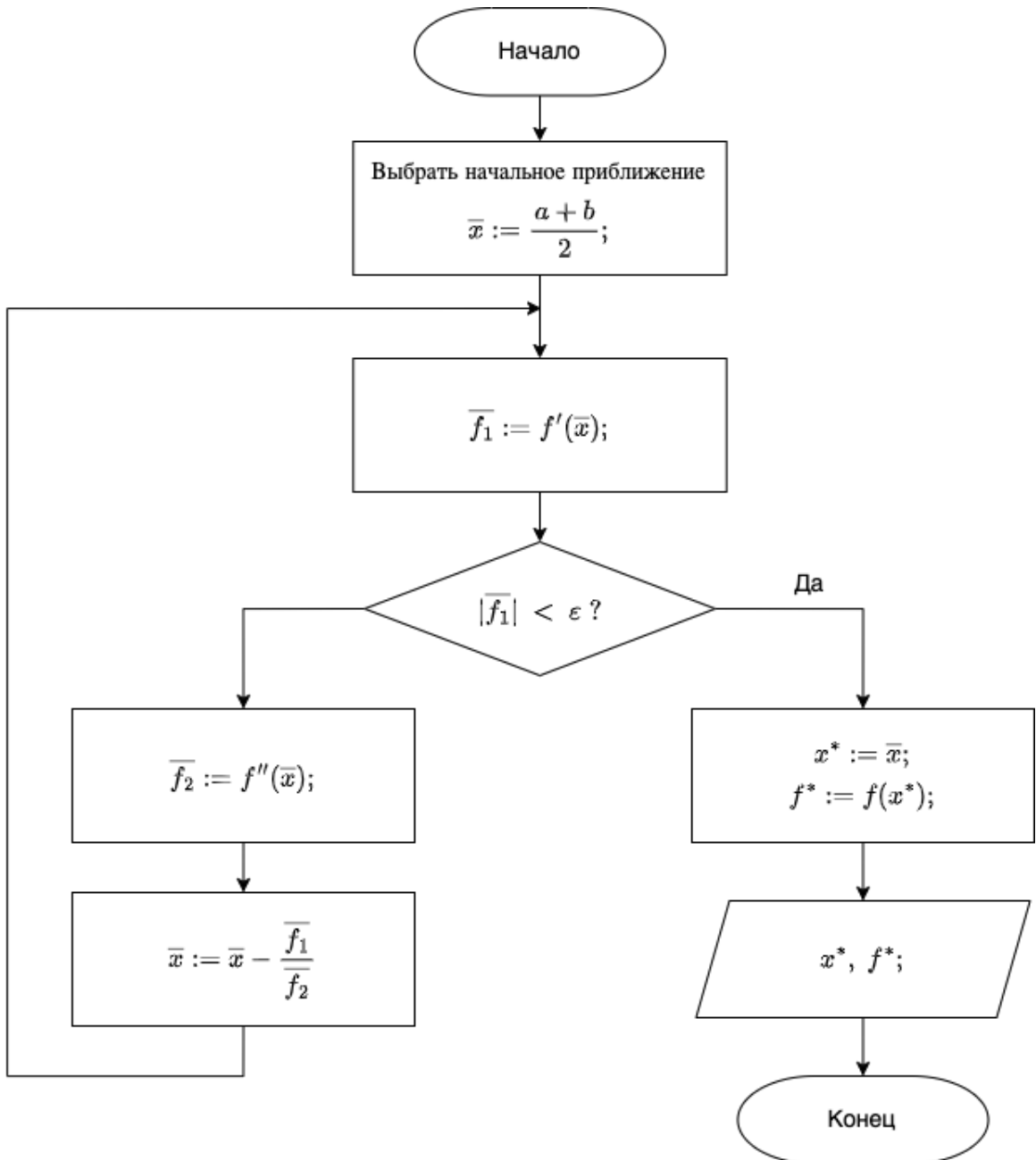


Рисунок 1.1 – Схема алгоритма метода Ньютона

2 Практическая часть

Таблица 2.1 – Результаты расчетов для задачи из индивидуального варианта

№ п/п	ε	N	x^*	$f(x^*)$
1	0.01	8	0.4826521305	-1.4738930849
2	0.0001	10	0.4824240968	-1.4738932842
3	0.000001	14	0.4824180859	-1.4738932844

Таблица 2.2 – Обобщающая таблица (для $\varepsilon = 1e - 6$)

№ п/п	Метод	N	x^*	$f(x^*)$
1	Поразрядного поиска	50	0.4824180603	-1.4738932844
2	Золотого сечения	31	0.4824184653	-1.4738932844
3	Парабол	11	0.4824179876	-1.4738932844
4	Ньютона модифицированный	14	0.4824180859	-1.4738932844
5	Функция fminbnd	9	0.4824181903	-1.4738932844

В листинге 2.1 представлен код программы.

Листинг 2.1 — Код программы

```
1  function lab_04()
2      clc();
3
4      a = 0;
5      b = 1;
6      delta = 1e-3;
7      eps = 1e-6;
8
9      debugFlag = true;
10     delay = 1;
11
12     fplot(@f, [a, b]);
13     hold on;
14
15     x = (a + b) / 2;
16     f2_x0 = (f(x - delta) - 2 * f(x) + f(x + delta)) / power(delta, 2);
17     N = 3;
18
19     while true
20         f1 = (f(x + delta) - f(x - delta)) / (2 * delta);
```

Продолжение листинга 2.1

```

21     N = N + 2;
22
23     if debugFlag
24         fprintf('N = %2d:   x = %.10f;   f1 = %.10f;\n\n', N, x, f1);
25         plot(x, f(x), 'xk');
26         pause(delay);
27     end
28
29     if abs(f1) < eps
30         break;
31     else
32         x = x - f1 / f2_x0;
33     end
34 end
35
36 x_star = x;
37 f_star = f(x);
38 N = N + 1;
39 scatter(x_star, f_star, 'filled');
40 fprintf('Ответ:   N = %2d;   x* = %.10f;   f(x*) = %.10f;\n\n', N, x_star,
↪ f_star);
41
42 options = optimset('TolX', eps);
43 if debugFlag
44     options = optimset(options, 'Display', 'iter');
45 end
46 [x_star, f_star] = fminbnd(@f, a, b, options);
47 fprintf('fminbnd:   x = %.10f;   f(x) = %.10f.\n\n', x_star, f_star);
48 end
49
50 function y = f(x)
51     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;
52 end

```