



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 № 4
по курсу «Методы вычислений»
на тему: «Метод Ньютона»
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

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

методом Ньютона с заданной точностью ε .

Входные данные

Заданная функция:

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

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 Ньютона

Схема метода Ньютона представлена на рисунке 1.

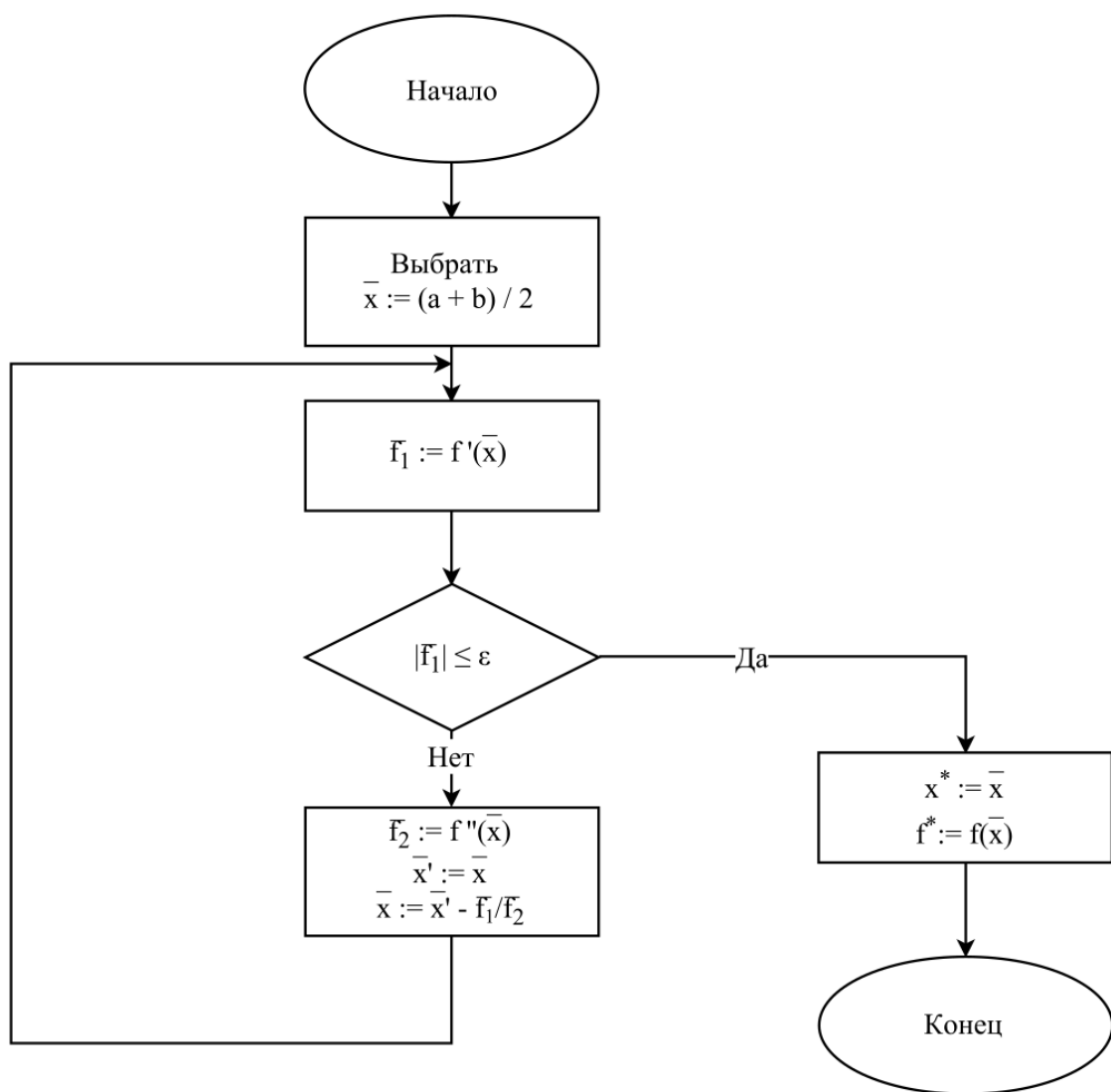


Рисунок 1 — Схема метода Ньютона

Результаты вычислений

Таблица 1 — Результаты вычислений методом Ньютона

№ п/п	ε	N	x^*	$f(x^*)$
1	10^{-2}	11	0,4774047325	−1,4738017976
2	10^{-4}	14	0,4823683107	−1,4738932752
3	10^{-6}	17	0,4824178114	−1,4738932844

Таблица 2 — Сводная таблица для сравнения методов, $\varepsilon = 10^{-6}$

№ п/п	Метод	N	x^*	$f(x^*)$
1	поразрядного поиска	50	0,4824180651	−1,4738932844
2	золотого сечения	31	0,4824184749	−1,4738932844
3	парабол	15	0,4824178751	−1,4738932844
4	Ньютона	17	0,4824178114	−1,4738932844
5	Функция fminbnd	10	0,4824181903	−1,4738932844

Текст программы

Листинг 1 — lab04.m

```

1 function lab04
2     clc;
3     debug = true;
4
5     a = 0;
6     b = 1;
7     eps = 1e-6;
8     dx = 1e-6;
9
10    fplot(@func func(x), [a, b], 'b');
11    hold on;
12
13    global N;
14    N = 0;
15
16    manual = true;
17    methods = ["Метод ньютона", "fminbnd"];
18    if (manual)
19        [x, f] = newton(a, b, eps, dx, debug);
20    else
21        [x, f] = fminbnd(@func, a, b, optimset('Display', 'iter', 'TolX', eps));
22    end
23    fprintf('Минимум функции: (x=%12.10f, f=%12.10f)\n', x, f);
24    fprintf('N = %d\n', N);
25    p = plot(x, f, 'rx', 'MarkerSize', 15);
26    legend(p, methods(1 + ~manual), 'Location', 'northwest');
27
28    hold off;
29 end
30
31 function y = func(x)
32     global N;
33     N = N + 1;
34
35     x3 = power(x, 3);
36     x2 = power(x, 2);
37     sqrt2 = sqrt(2);
38

```

```

39     ch = cosh((3 * x3 + 2 * x2 - 4 * x + 5) / 3);
40     th = tanh((x3 - 3 * sqrt2 * x - 2) / (2 * x + sqrt2));
41
42     y = ch + th - 2.5;
43 end
44
45 function [x, f] = newton(a, b, eps, dx, debug)
46     [x1, xr] = goldenSectionBoundaries(a, b);
47     x = (x1 + xr) / 2;
48
49     run = true;
50     iteration = 1;
51     while (run)
52         f_plus = func(x + dx);
53         f = func(x);
54         f_minus = func(x - dx);
55
56         if (debug)
57             fprintf('Итерация %d: [x=%12.10f, f=%12.10f]\n', iteration, x, f);
58             plot(x, f, 'g.', 'MarkerSize', 15);
59             iteration = iteration + 1;
60         end
61
62         x0 = x;
63
64         f1 = (f_plus - f) / dx;
65         f2 = (f_plus - 2 * f + f_minus) / (dx .^ 2);
66         x = x - f1/f2;
67
68         run = (abs(x - x0) > eps) || (abs(f1) > eps);
69     end
70
71     if (debug)
72         fprintf('Итерация %d: [x=%12.10f, f=%12.10f]\n', iteration, x, f);
73         plot(x, f, 'g.', 'MarkerSize', 15);
74     end
75 end
76
77 function [x1, xr] = goldenSectionBoundaries(a, b)
78     [~, ~, x1, xr] = goldenSectionSearch(a, b, 0.25, false);
79 end
80
81 function [x, f, x1, x2] = goldenSectionSearch(a, b, eps, debug)
82     tau = (sqrt(5) - 1) / 2;
83     delta = b - a;
84
85     x1 = b - tau * delta;
86     xr = a + tau * delta;
87     f1 = func(x1);
88     fr = func(xr);
89
90     iteration = 1;
91     while (delta > 2 * eps)
92         if (debug)
93             fprintf('Итерация %d: [a=%12.10f, b=%12.10f], (x1=%12.10f, xr=%12.10f)\n',
94                     iteration, a, b, x1, xr);
95             iteration = iteration + 1;
96         end
97
98         if (f1 > fr)
99             a = x1;

```

```

99         delta = b - a;
100        x1 = xr;
101        fl = fr;
102        xr = a + tau * delta;
103        fr = func(xr);
104    else
105        b = xr;
106        delta = b - a;
107        xr = x1;
108        fr = fl;
109        x1 = b - tau * delta;
110        fl = func(x1);
111    end
112 end
113
114 if (debug)
115     fprintf('Итерация %d: [a=%12.10f, b=%12.10f], (x1=%12.10f, xr=%12.10f)\n',
116           iteration, a, b, x1, xr);
117 end
118
119 x = (a + b) / 2;
120 f = func(x);
121
122 x1 = a;
123 x2 = b;
124 end

```