

МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)
ФАКУЛЬТЕТ АЭРОКОСМИЧЕСКИХ ТЕХНОЛОГИЙ

Лабораторная работа № 1
Численное решение нелинейных уравнений

Манулин Павел
Группа Б03-107

Сентябрь, 2023 г.

1 Постановка задачи

Локализовать корни уравнений:

$$3x^4 + 8x^3 + 6x^2 - 10 = 0$$

$$5 \ln x^2 + (x - 2)^3 = 0$$

А затем уточнить с помощью методов: половинного деления (дихотомии), метода простой итерации, метода Ньютона, модифицированного метода Ньютона, метода секущих с точностью 0,0001.

2 Метод нахождения отрезков локализации

Требуется найти непересекающиеся отрезки $[a_i, b_i]$, каждому из которых принадлежит один и только один корень данного уравнения x_i^* . Пусть

$$f(x) = a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n = 0,$$

Тогда требуется

1. Определить число этих корней $N = N_+ + N_-$

(а) Теорема Декарта: число положительных корней равно числу перемен знаков в последовательности коэффициентов a_0, a_1, \dots, a_n или на чётное число меньше.

(б) Все корни многочлена (включая комплексные) лежат в кольце

$$\frac{|a_n|}{|a_n| + B} \leq |z| \leq 1 + \frac{A}{|a_0|},$$

$$A = \max_{1 \leq i \leq n} |a_i|,$$

$$B = \max_{0 \leq i \leq n-1} |a_i|.$$

2. С помощью теоремы Декарта, Бюдана-Фурье, Штурма определить число действительных корней на каждом промежутке с учётом их кратности

Если уравнение состоит из трансцендентных функций, то локализуем корни графически.

3 Нахождение отрезков локализации для уравнений

Рассмотрим первое многочлен $p(x) = 3x^4 + 8x^3 + 6x^2 - 10$. Все корни данного многочлена будут лежать в кольце $\frac{5}{9} \leq |z| \leq \frac{39}{9}$. Далее воспользуемся теоремой Декарта:

$$3, 8, 6, -10 \Rightarrow 1 \text{ корень } x \in \mathbb{R}^+$$

$$x \rightarrow -x$$

$$3, -8, 6, -10 \Rightarrow \begin{bmatrix} 3 \text{ корня } x \in \mathbb{R}^- \\ 1 \text{ корень } x \in \mathbb{R}^- \end{bmatrix}$$

Для того, чтобы уточнить количество корней в промежутке $[-\frac{39}{9}, -\frac{5}{9}]$, воспользуемся теоремой Бюдано-Фурье:

Порядок производной	Функция	$a = -\frac{39}{9}$	$b = -\frac{5}{9}$
0	$p(x) = 3x^4 + 8x^3 + 6x^2 - 10$	>0	<0
1	$p'(x) = 12x^3 + 24x^2 + 12x$	<0	>0
2	$p''(x) = 36x^2 + 48x + 12$	>0	<0
3	$p'''(x) = 72x + 48$	<0	>0
4	$p''''(x) = 72$	>0	>0

$$\Delta = S(a+0) - S(b-0),$$

где $S(a+0)$ - число перемен знаков в последовательности функции и её производных в точке a , $S(b-0)$ - число перемен знаков в последовательности функции и её производных в точке b , Δ - число действительных корней с учётом их кратности на $[a, b]$ или меньше этого числа на чётное число.

Получим, что

$$\Delta = 4 - 3 = 1 \Rightarrow 1 \text{ корень } x \in [-\frac{39}{9}, -\frac{5}{9}]$$

$$1 \text{ корень } x \in [\frac{5}{9}, \frac{39}{9}]$$

Локализуем графически корни уравнения $5 \ln x^2 + (x-2)^3 = 0$

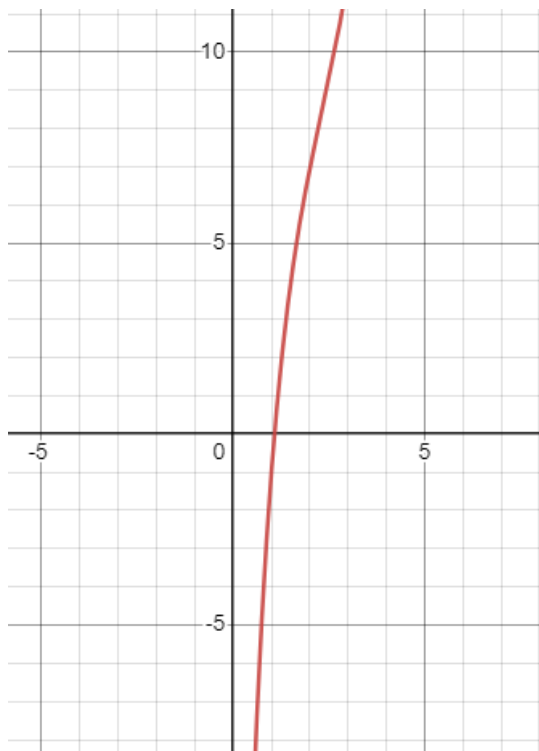


Рис. 1: График функции $5 \ln x^2 + (x-2)^3 = 0$

Далее будем рассматривать отрезок $[0, 5; 2]$

4 Теоретическое описание выбранных методов

1. **Метод половинного деления (дихотомия).** Отыскивается x - приближение к корню $x^* \in [a, b]$ с точностью ε .

Шаг 1. $m = 0$

Шаг 2. $a_m = a, b_m = b$

Шаг 3. $c = \frac{a_m + b_m}{2}$

Шаг 4.

- Если $f(c)f(a_m) > 0$, то $a_{m+1} = c, b_{m+1} = b_m$
- Если $f(c)f(b_m) > 0$, то $a_{m+1} = a_m, b_{m+1} = c$

Шаг 5. Если $|f(\frac{a_{m+1} + b_{m+1}}{2})| > \varepsilon$, то $m = m + 1$, перейти к Шагу 2,

иначе $x = \frac{a_{m+1} + b_{m+1}}{2}$

2. Метод простой итерации

$$f(x) = 0 \rightarrow x = g(x) \rightarrow x_{n+1} = g(x_n)$$

Условие сходимости: $\forall x \in [a, b] : |g'(x)| < 1$ или $\forall x', x'' \in [a, b] : |g(x') - g(x'')| \leq q|x' - x''|, q < 1$

3. Метод Ньютона (касательных)

$$f(x) = f(x_n) + f'(x_n)(x - x_n) \rightarrow x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Условие сходимости: $\frac{1}{2} \frac{M_2}{m_1} |x_0 - x^*|^2 < 1$

$$|x_{n+1} - x^*| < \frac{1}{2} \frac{M_2}{m_1} |x_n - x^*|^2 < C^{-1} (C|x_0 - x^*|)^{2^n}$$

Выбор начального приближения: $f(x_0) f''(x_0) > 0$.

Практический критерий оценки достижения заданной точности:

$$|x_{n+1} - x_*| < \frac{1}{2} \frac{M_2}{m_1} |x_{n+1} - x_n|^2 < \varepsilon.$$

4. Метод секущих В метод Ньютона подставим: $f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$

$$x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}$$

5. Метод простой итерации

$$f(x) = 0 \rightarrow x = g(x) \rightarrow x_{n+1} = g(x_n)$$

Условие сходимости:

$$|g(x) - g(y)| \leq q|x - y| \rightarrow |g(x_{n+1}) - g(x_*)| \leq q|x_n - x_*|, q = \max |g'(x)| < 1$$

Практические критерии оценки достижения заданной точности:

$$|x_{n+1} - x_n| < \varepsilon \text{ или } |f(x_n)| < \delta_\varepsilon$$

Если $g(x) = x - \frac{f(x)}{f'(x)}$, то получим метод Ньютона.

1 Таблицы расчетов

1.1 Дихотомия

	a	b	f(a)	f(b)	c	f(c)
0	-2,0759	-2,0758	0,0001	-0,0003	-2,0758	-0,0001
1	-2,0759	-2,0758	0,0006	-0,0003	-2,0759	0,0001
2	-2,0759	-2,0758	0,0006	-0,0011	-2,0758	-0,0003
3	-2,0759	-2,0758	0,0006	-0,0028	-2,0758	-0,0011
4	-2,0760	-2,0758	0,0039	-0,0028	-2,0759	0,0006
5	-2,0760	-2,0755	0,0039	-0,0094	-2,0758	-0,0028
6	-2,0764	-2,0755	0,0172	-0,0094	-2,0760	0,0039
7	-2,0774	-2,0755	0,0438	-0,0094	-2,0764	0,0172
8	-2,0792	-2,0755	0,0973	-0,0094	-2,0774	0,0438
9	-2,0829	-2,0755	0,2050	-0,0094	-2,0792	0,0973
10	-2,0903	-2,0755	0,4231	-0,0094	-2,0829	0,2050
11	-2,0903	-2,0608	0,4231	-0,4273	-2,0755	-0,0094
12	-2,0903	-2,0312	0,4231	-1,2202	-2,0608	-0,4273
13	-2,0903	-1,9722	0,4231	-2,6439	-2,0312	-1,2202
14	-2,2083	-1,9722	4,4524	-2,6439	-2,0903	0,4231
15	-2,4444	-1,9722	16,1143	-2,6439	-2,2083	4,4524
16	-2,4444	-1,5000	16,1143	-8,3125	-1,9722	-2,6439
17	-2,4444	-0,5556	16,1143	-9,2341	-1,5000	-8,3125
18	-4,3333	-0,5556	509,5185	-9,2341	-2,4444	16,1143

Рис. 1: Метод дихотомии для отрицательного корня $3x^4 + 8x^3 + 6x^2 - 10 = 0$

	a	b	f(a)	f(b)	c	f(c)
0	0,8260	0,8260	-0,0001	0,0001	0,8260	-0,0000
1	0,8260	0,8261	-0,0001	0,0003	0,8260	0,0001
2	0,8260	0,8261	-0,0006	0,0003	0,8260	-0,0001
3	0,8260	0,8261	-0,0006	0,0013	0,8261	0,0003
4	0,8260	0,8261	-0,0006	0,0032	0,8261	0,0013
5	0,8260	0,8263	-0,0006	0,0070	0,8261	0,0032
6	0,8258	0,8263	-0,0082	0,0070	0,8260	-0,0006
7	0,8258	0,8267	-0,0082	0,0223	0,8263	0,0070
8	0,8249	0,8267	-0,0387	0,0223	0,8258	-0,0082
9	0,8249	0,8286	-0,0387	0,0835	0,8267	0,0223
10	0,8212	0,8286	-0,1598	0,0835	0,8249	-0,0387
11	0,8212	0,8359	-0,1598	0,3308	0,8286	0,0835
12	0,8212	0,8507	-0,1598	0,8383	0,8359	0,3308
13	0,7917	0,8507	-1,0919	0,8383	0,8212	-0,1598
14	0,7917	0,9097	-1,0919	3,0434	0,8507	0,8383
15	0,7917	1,0278	-1,0919	8,3708	0,9097	3,0434
16	0,5556	1,0278	-6,4906	8,3708	0,7917	-1,0919
17	0,5556	1,5000	-6,4906	45,6875	1,0278	8,3708
18	0,5556	2,4444	-6,4906	249,8153	1,5000	45,6875
19	0,5556	4,3333	-6,4906	1.811,4444	2,4444	249,8153

Рис. 2: Метод дихотомии для положительного корня $3x^4 + 8x^3 + 6x^2 - 10 = 0$

	a	b	f(a)	f(b)	c	f(c)
0	1,0807	1,0808	-0,0006	0,0005	1,0808	-0,0001
1	1,0806	1,0808	-0,0017	0,0005	1,0807	-0,0006
2	1,0804	1,0808	-0,0038	0,0005	1,0806	-0,0017
3	1,0801	1,0808	-0,0082	0,0005	1,0804	-0,0038
4	1,0801	1,0815	-0,0082	0,0091	1,0808	0,0005
5	1,0801	1,0830	-0,0082	0,0263	1,0815	0,0091
6	1,0801	1,0859	-0,0082	0,0607	1,0830	0,0263
7	1,0742	1,0859	-0,0775	0,0607	1,0801	-0,0082
8	1,0625	1,0859	-0,2177	0,0607	1,0742	-0,0775
9	1,0625	1,1094	-0,2177	0,3315	1,0859	0,0607
10	1,0625	1,1562	-0,2177	0,8511	1,1094	0,3315
11	1,0625	1,2500	-0,2177	1,8096	1,1562	0,8511
12	0,8750	1,2500	-2,7591	1,8096	1,0625	-0,2177
13	0,5000	1,2500	-10,3065	1,8096	0,8750	-2,7591
14	0,5000	2,0000	-10,3065	6,9315	1,2500	1,8096

Рис. 3: Метод дихотомии для корня $5 \ln x^2 + (x - 2)^3 = 0$

1.2 Метод секущей

	a	b	f(a)	f(b)	c	f(c)
0	-4,3333	-2,0758	509,5185	-0,0001	-2,0758	-0,0001
1	-4,3333	-2,0758	509,5185	-0,0001	-2,0758	-0,0001
2	-4,3333	-2,0758	509,5185	-0,0001	-2,0758	-0,0001
3	-4,3333	-2,0758	509,5185	-0,0002	-2,0758	-0,0001
4	-4,3333	-2,0758	509,5185	-0,0002	-2,0758	-0,0002
5	-4,3333	-2,0758	509,5185	-0,0002	-2,0758	-0,0002
6	-4,3333	-2,0758	509,5185	-0,0003	-2,0758	-0,0002
7	-4,3333	-2,0758	509,5185	-0,0003	-2,0758	-0,0003
8	-4,3333	-2,0758	509,5185	-0,0003	-2,0758	-0,0003
9	-4,3333	-2,0758	509,5185	-0,0004	-2,0758	-0,0003
10	-4,3333	-2,0758	509,5185	-0,0004	-2,0758	-0,0004
11	-4,3333	-2,0758	509,5185	-0,0005	-2,0758	-0,0004
12	-4,3333	-2,0758	509,5185	-0,0006	-2,0758	-0,0005
13	-4,3333	-2,0758	509,5185	-0,0007	-2,0758	-0,0006
14	-4,3333	-2,0758	509,5185	-0,0008	-2,0758	-0,0007
15	-4,3333	-2,0758	509,5185	-0,0009	-2,0758	-0,0008
16	-4,3333	-2,0758	509,5185	-0,0010	-2,0758	-0,0009
17	-4,3333	-2,0758	509,5185	-0,0012	-2,0758	-0,0010
18	-4,3333	-2,0758	509,5185	-0,0013	-2,0758	-0,0012
19	-4,3333	-2,0758	509,5185	-0,0015	-2,0758	-0,0013
20	-4,3333	-2,0758	509,5185	-0,0017	-2,0758	-0,0015
21	-4,3333	-2,0758	509,5185	-0,0020	-2,0758	-0,0017
22	-4,3333	-2,0758	509,5185	-0,0023	-2,0758	-0,0020
23	-4,3333	-2,0758	509,5185	-0,0026	-2,0758	-0,0023
24	-4,3333	-2,0757	509,5185	-0,0030	-2,0758	-0,0026

Рис. 4: Метод секущей для отрицательного корня $3x^4 + 8x^3 + 6x^2 - 10 = 0$

В скриншот не поместилась вся таблица. Метод сошёлся за 108 итераций.

	a	b	f(a)	f(b)	c	f(c)
0	0,8260	4,3333	-0,0001	1.811,4444	0,8260	-0,0001
1	0,8260	4,3333	-0,0001	1.811,4444	0,8260	-0,0001
2	0,8260	4,3333	-0,0001	1.811,4444	0,8260	-0,0001
3	0,8260	4,3333	-0,0001	1.811,4444	0,8260	-0,0001
4	0,8260	4,3333	-0,0001	1.811,4444	0,8260	-0,0001
5	0,8260	4,3333	-0,0001	1.811,4444	0,8260	-0,0001
6	0,8260	4,3333	-0,0002	1.811,4444	0,8260	-0,0001
7	0,8260	4,3333	-0,0002	1.811,4444	0,8260	-0,0002
8	0,8260	4,3333	-0,0002	1.811,4444	0,8260	-0,0002
9	0,8260	4,3333	-0,0002	1.811,4444	0,8260	-0,0002
10	0,8260	4,3333	-0,0002	1.811,4444	0,8260	-0,0002
11	0,8260	4,3333	-0,0002	1.811,4444	0,8260	-0,0002
12	0,8260	4,3333	-0,0002	1.811,4444	0,8260	-0,0002
13	0,8260	4,3333	-0,0002	1.811,4444	0,8260	-0,0002
14	0,8260	4,3333	-0,0003	1.811,4444	0,8260	-0,0002
15	0,8260	4,3333	-0,0003	1.811,4444	0,8260	-0,0003
16	0,8260	4,3333	-0,0003	1.811,4444	0,8260	-0,0003
17	0,8260	4,3333	-0,0003	1.811,4444	0,8260	-0,0003
18	0,8260	4,3333	-0,0003	1.811,4444	0,8260	-0,0003
19	0,8260	4,3333	-0,0004	1.811,4444	0,8260	-0,0003
20	0,8260	4,3333	-0,0004	1.811,4444	0,8260	-0,0004
21	0,8260	4,3333	-0,0004	1.811,4444	0,8260	-0,0004
22	0,8260	4,3333	-0,0004	1.811,4444	0,8260	-0,0004
23	0,8260	4,3333	-0,0005	1.811,4444	0,8260	-0,0004

Рис. 5: Метод секущей для положительного корня $3x^4 + 8x^3 + 6x^2 - 10 = 0$

В скриншот не поместилась вся таблица. Метод сошёлся за 176 итераций.

	a	b	f(a)	f(b)	c	f(c)
0	0,5000	1,0808	-10,3065	0,0002	1,0808	0,0001
1	0,5000	1,0808	-10,3065	0,0006	1,0808	0,0002
2	0,5000	1,0809	-10,3065	0,0018	1,0808	0,0006
3	0,5000	1,0812	-10,3065	0,0054	1,0809	0,0018
4	0,5000	1,0821	-10,3065	0,0162	1,0812	0,0054
5	0,5000	1,0849	-10,3065	0,0482	1,0821	0,0162
6	0,5000	1,0930	-10,3065	0,1427	1,0849	0,0482
7	0,5000	1,1170	-10,3065	0,4183	1,0930	0,1427
8	0,5000	1,1883	-10,3065	1,1904	1,1170	0,4183
9	0,5000	1,3968	-10,3065	3,1227	1,1883	1,1904
10	0,5000	2,0000	-10,3065	6,9315	1,3968	3,1227

Рис. 6: Метод секущей для корня $5 \ln x^2 + (x - 2)^3 = 0$

1.3 Метод Ньютона

	x	fun(x)	fun``(x)
0	-2,0758	0,0000	67,4884
1	-2,0760	0,0043	67,5035
2	-2,0872	0,3319	68,6455
3	-2,1811	3,4302	78,5713
4	-2,4444	16,1143	109,7778

Рис. 7: Метод Ньютона для отрицательного корня $3x^4 + 8x^3 + 6x^2 - 10 = 0$

	x	fun(x)	fun``(x)
0	0,8260	0,0000	76,2144
1	0,8263	0,0082	76,2411
2	0,8409	0,4983	77,8153
3	0,9486	4,6581	89,9303
4	1,2350	21,1986	126,1865
5	1,7266	75,7294	202,2013
6	2,4444	249,8153	344,4444

Рис. 8: Метод Ньютона для положительного корня $3x^4 + 8x^3 + 6x^2 - 10 = 0$

	x	fun(x)	fun`'(x)
0	1,0808	0,0001	-14,0764
1	1,0808	-0,0001	-14,0768
2	1,0808	0,0001	-14,0763
3	1,0808	-0,0002	-14,0769
4	1,0808	0,0002	-14,0762
5	1,0807	-0,0002	-14,0770
6	1,0808	0,0003	-14,0760
7	1,0807	-0,0004	-14,0773
8	1,0808	0,0005	-14,0757
9	1,0807	-0,0006	-14,0776
10	1,0808	0,0007	-14,0753
11	1,0807	-0,0009	-14,0781
12	1,0809	0,0010	-14,0746
13	1,0807	-0,0013	-14,0789
14	1,0809	0,0016	-14,0736
15	1,0806	-0,0019	-14,0801
16	1,0810	0,0024	-14,0722
17	1,0805	-0,0029	-14,0819
18	1,0811	0,0036	-14,0700
19	1,0804	-0,0044	-14,0847
20	1,0812	0,0054	-14,0666
21	1,0802	-0,0066	-14,0888
22	1,0815	0,0081	-14,0616
23	1,0799	-0,0099	-14,0950
24	1,0818	0,0122	-14,0540

В скриншот не поместилась вся таблица. Метод сошёлся за 46 итераций.

Рис. 9: Метод Ньютона для корня $5 \ln x^2 + (x - 2)^3 = 0$

1.4 Модифицированный метод Ньютона

	x	fun(x)	fun``(x)
0	-2,0759	0,0001	67,4888
1	-2,0759	0,0002	67,4890
2	-2,0759	0,0003	67,4896
3	-2,0759	0,0006	67,4906
4	-2,0759	0,0012	67,4925
5	-2,0759	0,0022	67,4961
6	-2,0760	0,0041	67,5030
7	-2,0761	0,0078	67,5159
8	-2,0764	0,0148	67,5404
9	-2,0768	0,0280	67,5868
10	-2,0777	0,0531	67,6749
11	-2,0793	0,1009	67,8424
12	-2,0825	0,1928	68,1631
13	-2,0886	0,3720	68,7840
14	-2,1005	0,7323	70,0142
15	-2,1251	1,5039	72,5727
16	-2,1811	3,4302	78,5713
17	-2,4444	16,1143	109,7778

Рис. 10: Модифицированный метод Ньютона для отрицательного корня $3x^4 + 8x^3 + 6x^2 - 10 = 0$

	x	fun(x)	fun``(x)
0	0,8260	0,0001	76,2147
1	0,8260	0,0001	76,2147
2	0,8260	0,0001	76,2147
3	0,8260	0,0001	76,2148
4	0,8260	0,0001	76,2148
5	0,8260	0,0002	76,2149
6	0,8260	0,0002	76,2149
7	0,8260	0,0002	76,2150
8	0,8260	0,0002	76,2150
9	0,8260	0,0002	76,2151
10	0,8260	0,0003	76,2152
11	0,8260	0,0003	76,2153
12	0,8261	0,0003	76,2154
13	0,8261	0,0003	76,2155
14	0,8261	0,0004	76,2156
15	0,8261	0,0004	76,2157
16	0,8261	0,0005	76,2159
17	0,8261	0,0005	76,2160
18	0,8261	0,0006	76,2162
19	0,8261	0,0006	76,2164
20	0,8261	0,0007	76,2166
21	0,8261	0,0008	76,2169
22	0,8261	0,0008	76,2171
23	0,8261	0,0009	76,2174
24	0,8261	0,0010	76,2177

Рис. 11: Модифицированный метод Ньютона для положительного корня $3x^4 + 8x^3 + 6x^2 - 10 = 0$

	x	fun(x)	fun`'(x)
0	1,0808	0,0001	-14,0764
1	1,0808	-0,0001	-14,0768
2	1,0808	0,0002	-14,0763
3	1,0807	-0,0002	-14,0770
4	1,0808	0,0003	-14,0759
5	1,0807	-0,0005	-14,0775
6	1,0808	0,0007	-14,0752
7	1,0807	-0,0011	-14,0786
8	1,0809	0,0016	-14,0736
9	1,0806	-0,0023	-14,0808
10	1,0811	0,0034	-14,0703
11	1,0804	-0,0049	-14,0857
12	1,0814	0,0072	-14,0632
13	1,0799	-0,0105	-14,0961
14	1,0821	0,0154	-14,0480
15	1,0789	-0,0225	-14,1183
16	1,0836	0,0330	-14,0156
17	1,0767	-0,0481	-14,1658
18	1,0868	0,0705	-13,9462
19	1,0721	-0,1027	-14,2675
20	1,0937	0,1511	-13,7981
21	1,0624	-0,2191	-14,4857
22	1,1088	0,3248	-13,4814
23	1,0421	-0,4671	-14,9567
24	1,1429	0,7056	-12,7991

Рис. 12: Модифицированный метод Ньютона для корня $5 \ln x^2 + (x - 2)^3 = 0$

1.5 Метод простой итерации

	x	fun(x)	phi'(x)
0	1,0808	0,0000	-0,2980
1	1,0808	-0,0002	-0,2980
2	1,0808	0,0006	-0,2980
3	1,0806	-0,0021	-0,2981
4	1,0814	0,0077	-0,2978
5	1,0784	-0,0280	-0,2990
6	1,0895	0,1026	-0,2946
7	1,0500	-0,3695	-0,3105

Рис. 13: Метод простой итерации для $5 \ln x^2 + (x - 2)^3 = 0$

Для уравнения $3x^4 + 8x^3 + 6x^2 - 10 = 0$ при любом выборе $\phi(x)$ метод не сходится для обеих корней.

2 Оценка погрешности

	x*	метод дихотомии	метод секущей	метод ньютона	модиф. метод ньютона	метод простой итерации
0	0,8260	0,0000	0,0000	0,0000	0,0000	-
1	-2,0758	0,0000	0,0000	0,0000	0,0001	-
2	1,0807	-0,0001	-0,0001	-0,0001	-0,0001	0,0001

Рис. 14: Оценка абсолютных погрешностей для всех методов

3 Код

```
[2] import math
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

from matplotlib import style
```

```
[102] def fun(x):
    if x != 0:
        return 5 * math.log(x**2, math.e) + (x - 2)**3
    else:
        print("math domain error!")

def polinom_fun(x):
    return 3 * x**4 + 8 * x**3 + 6 * x**2 - 10

def diff_polinom_fun(x):
    return 12 * x**3 + 24 * x**2 + 12 * x

def diff2_polinom_fun(x):
    return 36 * x**2 + 48 * x + 12

def diff_fun(x):
    return 10/x + 3*(x - 2)

def diff2_fun(x):
    return 6 * (x - 2) - 10/(x**2)

def phi_fun(x):
    if x != 0:
        return 2 - (5 * math.log(x**2, math.e))**(1/3)
    else:
        print("This point does not require the scope of definition")

def phi_polinom_fun(x):
    if -1/3 * (10 - 6 * x**2 - 8 * x**3) >= 0:
        return (-1/3 * (10 - 6 * x**2 - 8 * x**3))**(1/4)

def diff_phi_polinom_fun(x):
    if (((10/3 - 2 * x**2 - 8/3 * x**3)**(3/4)) != 0) & (
        10/3 - 2 * x**2 - 8/3 * x**3 >= 0):
        return (x + 2 * x**2) / ((10/3 - 2 * x**2 - 8/3 * x**3)**(3/4))
```

```

[102] def diff_phi_fun(x):
    if (x != 0) & ((3 * x * (5 * np.log([x ** 2]))**(2/3)) != 0):
        return 2 - 10 / (3 * x * (5 * np.log([x ** 2]))**(2/3))
    else:
        print("This point does not require the scope of definition")

def make_pretty(df):
    df = df.style \
        .format(precision=4, thousands=".", decimal=",") \
        .format_index(str.lower, axis=1)
    return df

[112] def dichotomy_method(e, a, b, fun):
    df = pd.DataFrame(columns = ['a', 'b', 'f(a)', 'f(b)', 'c', 'f(c)'])
    x = (a + b)/2
    df.loc[-1] = [a, b, fun(a), fun(b), x, fun(x)]
    df.index = df.index + 1
    df = df.sort_index()

    while abs(fun(x)) > e:
        if fun(a) * fun(x) < 0:
            b = x
            x = (a + b) / 2
        elif fun(x) * fun(b) < 0:
            a = x
            x = (a + b) / 2
        df.loc[-1] = [a, b, fun(a), fun(b), x, fun(x)]
        df.index = df.index + 1
        df = df.sort_index()
    return(make_pretty(df))

def secant_method(e, a, b, fun):
    x = a + (fun(a) * (a - b))/(fun(b) - fun(a))
    df = pd.DataFrame(columns = ['a', 'b', 'f(a)', 'f(b)', 'c', 'f(c)'])
    df.loc[-1] = [a, b, fun(a), fun(b), x, fun(x)]
    df.index = df.index + 1
    df = df.sort_index()

    while abs(fun(x)) > e:
        if fun(a) * fun(x) < 0:
            b = x
            x = a + (fun(a) * (a - b))/(fun(b) - fun(a))
        elif fun(b) * fun(x) < 0:

```

```

[112]         elif fun(b) * fun(x) < 0:
            a = x
            x = a + (fun(a) * (a - b)) / (fun(b) - fun(a))
            df.loc[-1] = [a, b, fun(a), fun(b), x, fun(x)]
            df.index = df.index + 1
            df = df.sort_index()
        return(make_pretty(df))

def tangent_method(e, x_0, fun, diff_fun, diff2_fun):
    df = pd.DataFrame(columns = ['x', 'fun(x)', 'fun``(x)'])
    df.loc[-1] = [x_0, fun(x_0), diff2_fun(x_0)]
    df.index = df.index + 1
    df = df.sort_index()
    #conditions for a point from the localization segment: f(x)f'(x) > 0
    if diff2_fun(x_0) * fun(x_0) > 0:
        while abs(fun(x_0)) > e:
            x_0 = x_0 - (fun(x_0)) / (diff_fun(x_0))
            df.loc[-1] = [x_0, fun(x_0), diff2_fun(x_0)]
            df.index = df.index + 1
            df = df.sort_index()
        return(make_pretty(df))
    else:
        return("Initial point does not require conditions,"
              "try to enter another one")

def modified_tangent_method(e, x_0, fun, diff_fun, diff2_fun):
    #conditions for a point from the localization segment: f(x)f'(x) > 0
    df = pd.DataFrame(columns = ['x', 'fun(x)', 'fun``(x)'])
    df.loc[-1] = [x_0, fun(x_0), diff2_fun(x_0)]
    df.index = df.index + 1
    df = df.sort_index()

```

```

[112] def simple_iteration_method(e, x_0, fun, phi_fun, diff_phi_fun):
    #conditions for a point from the localization segment: |phi'(x_0)| < 1
    df = pd.DataFrame(columns = ['x', 'fun(x)', 'phi'(x)'])
    df.loc[-1] = [x_0, fun(x_0), diff_phi_fun(x_0)]
    df.index = df.index + 1
    df = df.sort_index()

    if abs(diff_phi_fun(x_0)) < 1:
        while abs(fun(x_0)) > e:
            x_0 = phi_fun(x_0)
            df.loc[-1] = [x_0, fun(x_0), diff_phi_fun(x_0)]
            df.index = df.index + 1
            df = df.sort_index()
        return(make_pretty(df))
    else:
        return("Initial point does not require conditions,"
               "try to enter another one")

def build_graph(a, b, fun):
    fig, ax = plt.subplots()
    ax.set_title('График функции')
    ax.set_xlabel('x')
    ax.set_ylabel('f(x)')
    y = []
    x = np.linspace(a, b, 1000)[:,:]
    for elem in x:
        y.append(fun(elem))
    ax.plot(x, y, 'r')
    plt.show()

```

4 Вывод

Мы показали, что все методы достаточно точно приближают корни данных уравнений с точностью до $\varepsilon = 0.0001$. Вопрос состоит в скорости сходимости. Самыми быстрыми методами оказались: Метод Ньютона и Метод простой итерации. Но МПИ работает не всегда. В данном случае, МПИ сходится только для корня уравнения $5 \ln x^2 + (x - 2)^3 = 0$, причём делает он это довольно быстро. Метод дихотомии оказался чуть медленнее, чем Метод Ньютона и МПИ, но быстрее остальных.