Федеральное государственное автономное образовательное учреждение высшего образования «Московский физико-технический институт (национальный исследовательский университет)»

ФАКУЛЬТЕТ АЭРОФИЗИКИ И КОСМИЧЕСКИХ ТЕХНОЛОГИЙ

Группа
Б03-908

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ - 1

Численное решение нелинейных уравнений

Выполнил:

Агеев Рамиль Наильевич

/ / /
(подпись) *(дата)*

Долгопрудный
2021г.

Содержание

1 Постановка задачи

Для двух уравнений

$$f(x) = 2x^5 - 5x^4 + 15x^2 - 7 = 0 \tag{1}$$

$$f(x) = x + ln(x) = 0 \tag{2}$$

1. Локализовать корни уравнения (найти непересекающиеся отрезки, каждый из которых имеет только один корень).

2. Формулы выбранных методов уточнения корней с обоснованием их сходимости.

3. Таблицы расчетных данных.

4. Оценка погрешности найденного решения (сравнение найденного решения с аналитическим решением).

2 Методы исследования корней функций

2.1 Локализация корней

Для начала определим, что подразумевается под локализацией.
В общем случае, рассмотрим некоторый полином n-й степени:

$$f(z) = a_0 z^n + a_1 z^{n-1} + ... + a_{n-1} z + a_n = 0, a_i \in R \tag{3}$$

И, благодаря следствию из основной теоремы алгебры, мы знаем, что все корни $3 \in R$:

$$\frac{|a_n|}{|a_n| + B} \leq |z| \leq 1 + \frac{A}{a_0}, \tag{4}$$

$$A = max|a_i|, (1 \leq i \leq n)$$
$$\mathrm{B} = \mathrm{max}|\mathrm{b}_i|, (0 \leq i \leq n-1)$$

Здесь и далее имеем следующее условие отыскания корня: Для каждого корня $x_i^* \in [a_i, b_i]$ уравнения требуется найти $\tilde{x}_i$ такое, что $|\tilde{x}_i - x_i^*| < \varepsilon$, где ε - заданная погрешность равная10^{-4} и $i \in [1, 2, 3]$.

2.2 Метод Деления Отрезка Пополам

Ищем $\tilde{x}$ - приближение к корню $x^* \in [a, b]$ с точностью ε.

Шаг 1-й $m = 0$

Шаг 2-й $a_m = a, b_m = b$

Шаг 3-й $c = \frac{a_m + b_m}{2}$

Шаг 4-й (а) Если $f(c)f(a_m) > 0$, то $a_{m+1} = c, b_{m+1} = b_m$ (б) Если $f(c)f(b_m) > 0$, то $a_{m+1} = a_m, b_{m+1} = c$

Шаг 5-й Если $|b_{m+1} - a_{m+1}| > \varepsilon$, то $m = m + 1$, перейти к Шагу 2, иначе $\tilde{x} = \frac{a_{m+1} + b_{m+1}}{2}$

2.3 Метод простой итерации

$$f(x) = 0 \to x = g(x) \to x_{n+1} = g(x_n) \tag{5}$$

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

2.4 Метод Ньютона

$$f(x) = f(x_n) + f'(x_n)(x - x_n) \to x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{6}$$

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:

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

$M_2 = max|f''(\xi)|$ на $[a, b]$
$m_1 = min|f''(\xi)|$ на $[a, b]$

2.5 Модифицированный Метод Ньютона

Формула метода имеет такой же вид, что и в (6). Однако, модификация заключается в том, что при выполнении задаваемого нами условия минимальной необходимой точности(например, отклонения не на 10 в -4, а на 10 в -3), мы можем заменить в формуле $f'(x_n)$ на $f'(x_0)$, чтобы упростить дальнейшие вычисления.

2.6 Метод Секущих

В методе Ньютона подставим: $f'(x_n) \approx \dfrac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$

$$x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})} \tag{7}$$

3 Исследование корней функций

3.1 Первая функция

Теперь перейдём к использованию методов для нахождения корней данного уравнения (1). Здесь и далее мы рассматриваем функцию (1). Для начала узнаем как выглядит график данной функции.

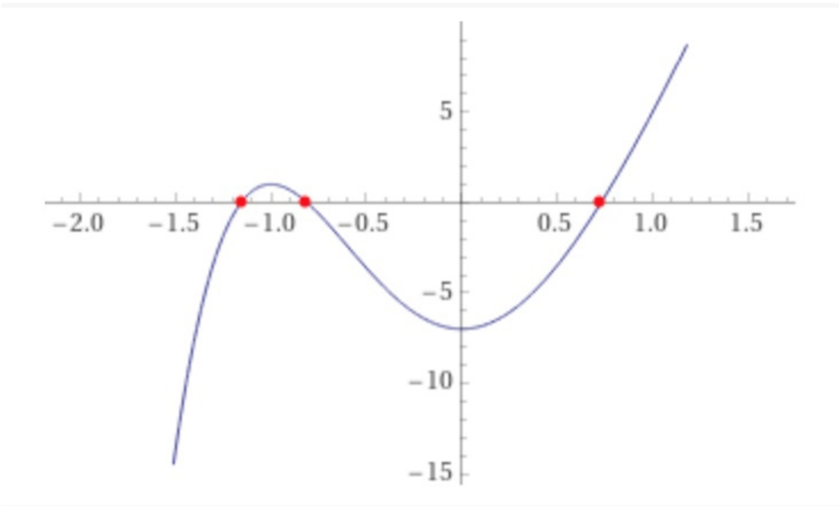

Рис. 1: График функции (1) (wolfram)

Видно, что можно выделить 3 отрезка локализации [a, b] (слева направо):

a) [-1.5, -1.0]

b) [-1.0, -0.5]

c) [0.5, 1.0]

3.2 Метод Деления Отрезка Пополам

Код:

```python
def bisection(a, b, e, f_A, f_B, f_X, X, f):
    x_ = (a + b) / 2
    i = 1
    while abs(a - b) > e:
        i += 1
        x_ = (a + b) / 2
        X.append(x_)
        f_a = f.subs(x, a)
        f_A.append(f_a)
        f_b = f.subs(x, b)
        f_B.append(f_b)
        f_x_ = f.subs(x, x_)
        f_X.append(f_x_)
        if f_a * f_x_ < 0:
            b = x_
        else:
            a = x_
    return x_, f_A, f_B, f_X, X
```

На отрезке a) за 13 итераций получаем $\tilde{x} = -1.15509033203125$ со следующей таблицей вычислений:

Iteration	x_	f(a)	f(b)	f((a+b)/2)
1	-1.25	-13.7500000000000	1.00000000000000	-1.87304687500000
2	-1.125	-1.87304687500000	1.00000000000000	0.371276855468750
3	-1.1875	-1.87304687500000	0.371276855468750	-0.513143539428711
4	-1.15625	-0.513143539428711	0.371276855468750	-0.0162070393562317
5	-1.140625	-0.0162070393562317	0.371276855468750	0.190648777410388
6	-1.1484375	-0.0162070393562317	0.190648777410388	0.0905693767708726
7	-1.15234375	-0.0162070393562317	0.0905693767708726	0.0380271311023535
8	-1.154296875	-0.0162070393562317	0.0380271311023535	0.0111226459476370
9	-1.1552734375	-0.0162070393562317	0.0111226459476370	-0.00248890768540200
10	-1.15478515625	-0.00248890768540200	0.0111226459476370	0.00433017399670277
11	-1.155029296875	-0.00248890768540200	0.00433017399670277	0.000923961544806673
12	-1.1551513671875	-0.00248890768540200	0.000923961544806673	-0.000781640701337238
13	-1.15509033203125	-0.000781640701337238	0.000923961544806673	0.0000713684800128789

Рис. 2: Таблица a) для метода бисекции

На отрезке b) за 13 итераций получаем $\tilde{x} = -0.81085205078125$ со следующей таблицей вычислений:

```
Iteration | x_ | f(a) | f(b) | f((a+b)/2)
1 | -0.75 | 1.00000000000000 | -3.62500000000000 | -0.619140625000000
2 | -0.875 | 1.00000000000000 | -0.619140625000000 | 0.527648925781250
3 | -0.8125 | 0.527648925781250 | -0.619140625000000 | 0.0151271820068359
4 | -0.78125 | 0.0151271820068359 | -0.619140625000000 | -0.289448320865631
5 | -0.796875 | 0.0151271820068359 | -0.289448320865631 | -0.133700137957931
6 | -0.8046875 | 0.0151271820068359 | -0.133700137957931 | -0.0583802577457391
7 | -0.80859375 | 0.0151271820068359 | -0.0583802577457391 | -0.0213947801657923
8 | -0.810546875 | 0.0151271820068359 | -0.0213947801657923 | -0.00307520532982153
9 | -0.8115234375 | 0.0151271820068359 | -0.00307520532982153 | 0.00604071881650903
10 | -0.81103515625 | 0.00604071881650903 | -0.00307520532982153 | 0.00148642909659369
11 | -0.810791015625 | 0.00148642909659369 | -0.00307520532982153 | -0.000793471310604588
12 | -0.8109130859375 | 0.00148642909659369 | -0.000793471310604588 | 0.000346708254741479
13 | -0.81085205078125 | 0.000346708254741479 | -0.000793471310604588 | -0.000223324207528464
```

Рис. 3: Таблица b) для метода бисекции

На отрезке c) за 13 итераций получаем $\tilde{x} = 0.73065185546875$ со следующей таблицей вычислений:

```
Iteration | x_ | f(a) | f(b) | f((a+b)/2)
1 | 0.75 | -3.50000000000000 | 5.00000000000000 | 0.330078125000000
2 | 0.625 | -3.50000000000000 | 0.330078125000000 | -1.71282958984375
3 | 0.6875 | -1.71282958984375 | 0.330078125000000 | -0.719995498657227
4 | 0.71875 | -0.719995498657227 | 0.330078125000000 | -0.201726496219635
5 | 0.734375 | -0.201726496219635 | 0.330078125000000 | 0.0625297110527754
6 | 0.7265625 | -0.201726496219635 | 0.0625297110527754 | -0.0700155246886425
7 | 0.73046875 | -0.0700155246886425 | 0.0625297110527754 | -0.00384648095678131
8 | 0.732421875 | -0.00384648095678131 | 0.0625297110527754 | 0.0293158094258956
9 | 0.7314453125 | -0.00384648095678131 | 0.0293158094258956 | 0.0127282018714769
10 | 0.73095703125 | -0.00384648095678131 | 0.0127282018714769 | 0.00443924349411517
11 | 0.730712890625 | -0.00384648095678131 | 0.00443924349411517 | 0.000295976856127922
12 | 0.7305908203125 | -0.00384648095678131 | 0.000295976856127922 | -0.00177535317493871
13 | 0.73065185546875 | -0.00177535317493871 | 0.000295976856127922 | -0.000739713437872824
```

Рис. 4: Таблица c) для метода бисекции

3.2.1 Вычисление погрешностей

Относительная погрешность:

$$\delta = \frac{|(x^* - \tilde{x})|}{x^*} * 100\% \tag{8}$$

Абсолютная погрешность:

$$d = (|x^* - \tilde{x}|) \tag{9}$$

В случае a) при $x^* = -1.1550954397842009827768773$ получаем $\delta_a = 0.044219315349329876\%$, и $d_a = 5.107752951039046e - 06$, что входит в допустимый диапозон значений.

В случае b) при $x^* = -0.810875961339023190673846667$ получаем $\delta_b = 0.00294873185458508 98\%$, и $d_b = 2.3910557773176855e-05$, что входит в допустимый диапозон значений.

В случае c) при $x^* = 0.730695448462151710365385 81$ получаем $\delta_c = 0.00596595934646621 3\%$, и $d_c = 4.359299340173095e-05$, что входит в допустимый диапозон значений.

3.3 Метод простой итерации

```python
def msi(x_0, X_0, X_1, phi, e):
    i = 0
    x_1 = phi.diff(x).subs(x, x_0)
    if abs(x_1) < 1.:
        while abs(x_0 - x_1) > e:
            i += 1
            X_1.append(x_1)
            X_0.append(x_0)
            if i == 1:
                print("Iteration", "|", "x_n+1 = phi(x_n)", "|", "x_n")
            print(f"{i}", "|", f"{x_1}", "|", f"{x_0}")
            x_0 = x_1
            x_1 = phi.subs(x, x_1)
    return x_0, X_0, X_1
```

В данном случае посчитать все три приближенных значения не получилось, и далее объясним почему. Однако отметим, что в случае c) получили $\tilde{x} = 0.730582799133599$ со следующей таблицей вычислений при $x_0 = -0.5$: При $\phi(x) = \sqrt{\frac{-2x^5+5x^4+7}{15}}$

```
Iteration | x_n+1 = phi(x_n) | x_n
1 | -0.148557090169228 | -0.5
2 | 0.683255928594648 | -0.148557090169228
3 | 0.720734642227232 | 0.683255928594648
4 | 0.728478938046713 | 0.720734642227232
5 | 0.730196406177608 | 0.728478938046713
6 | 0.730582799133599 | 0.730196406177608
```

Рис. 5: Таблица случая c) МПИ

При данном выборе ϕ мы каждый раз сходимся к одному и тому же значению. При выборе других: $\phi = \sqrt[4]{(2x^5 + 15x^2 - 7)/5}$ получаем для производной данной функции

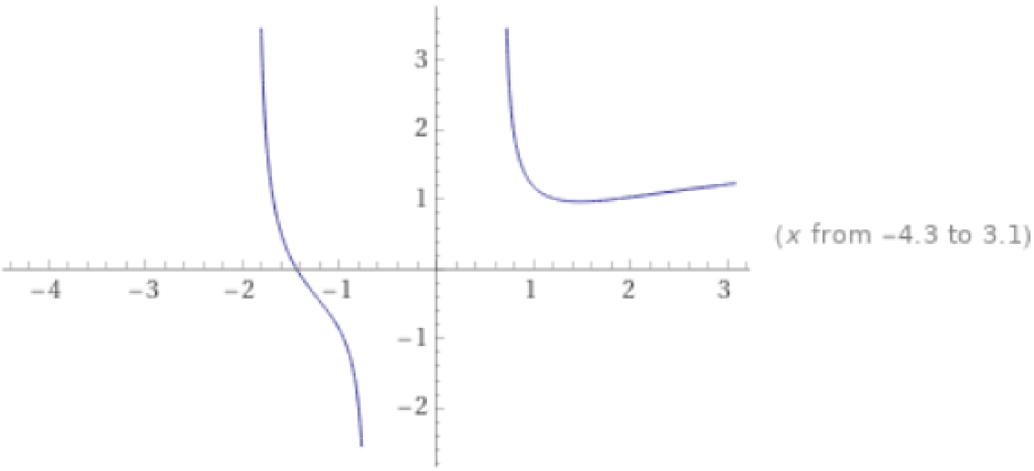

Рис. 6: График производной ϕ_2

И имеем, что для нее при некоторых $x_0 \leq 0$ значение модуля производной будет меньше 1, но функция сходиться ни к чему не будет(можно проверить при подстановке в код своих значений). При $\phi = \sqrt[5]{(5x^4 - 15x^2 + 7)/2}$ И по той же причине не подходит.

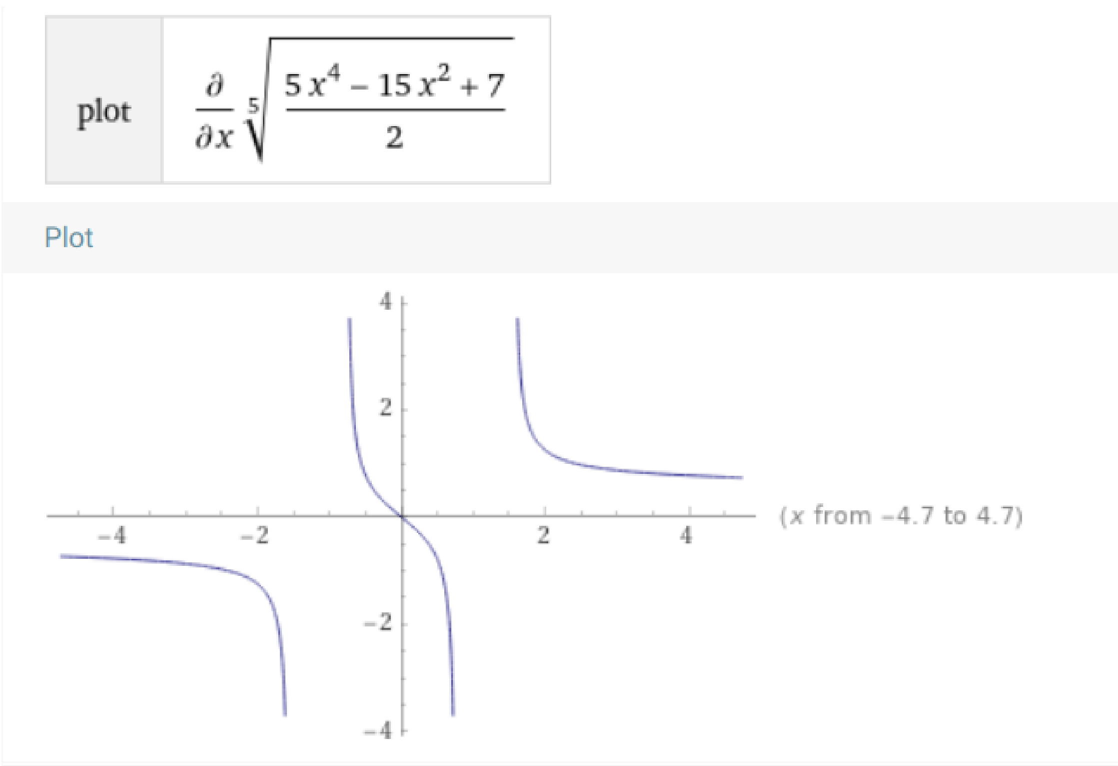

Рис. 7: График производной ϕ_3

3.3.1 Вычисление погрешностей

Посчитаем погрешности хотя бы для случая c): $\delta_c = 0.015416727829602763\%$, и $d_c = 0.000112649328552$

3.4 Метод Ньютона

```python
def newton(x_1, x_0, f, f_, X_0, F, F_):
    while abs(x_1.subs(x, x_0) - x_0) > e:
        X_0.append(x_0)
        F.append(f.subs(x, x_0))
        F_.append(f_.subs(x, x_0))
        x_0 = x_1.subs(x, x_0)
        x_1 = x_0 - f / f_
    return x_0, X_0, F, F_
```

Ниже приведены таблицы для трех различных x_0 и получились соответствующие корни -1.155095533695

```
F(x0)*F''(x0) = 93.4672128000000
x0 = -1.2
Iteration | x0 | f(x0) | f'(x_0)
1 | -1.2 | -0.744640000000000 | 19.2960000000000
2 | -1.16140961857380 | -0.0904664354595450 | 14.6841416350360
3 | -1.15524879299012 | -0.00214409712220487 | 13.9899973239884
-1.15509553369545
F(x0)*F''(x0) = 15.5687168000000
x0 = -0.7
Iteration | x0 | f(x0) | f'(x_0)
1 | -0.7 | -1.18664000000000 | -11.7390000000000
2 | -0.801085271317829 | -0.0928889824478163 | -9.63255553474314
3 | -0.810728504841143 | -0.00137752146919157 | -9.34415172449000
-0.810875925549818
F(x0)*F''(x0) = 70.0000000000000
x0 = 0.5
Iteration | x0 | f(x0) | f'(x_0)
1 | 0.5 | -3.50000000000000 | 13.1250000000000
2 | 0.766666666666667 | 0.618993251028808 | 17.4422345679012
3 | 0.731178481444807 | 0.00819813419430093 | 16.9754793411857
0.730695541668584
```

Рис. 8: Таблицы Метода Ньютона

3.4.1 Вычисление погрешностей

Считаем погрешности:
$\delta_a = 8.130172258063333e-06$, $d_a = 9.391124899948977e-08$, $\delta_b = 4.413647322324443e-06$, $d_b = 3.578920515501238e-08$, $\delta_c = 1.2755852309624322e-05, d_c = 9.320643223897918e-08$

3.5 Модифицированный Метод Ньютона

Этот метод практически такой же(если говорить о x_0 и условии выбора x_0), поэтому прилагаю таблицы и код.

```
def mod_newton(x_1, x_0, f, f_):
    i = 0
    x_tmp = x_0
    flag = True
    while abs(x_1.subs(x, x_0) - x_0) > e:
        i+=1
        if abs(x_1.subs(x, x_0) - x_0) < e + 0.006:
            flag = False
        if i ==1:
            print("Iteration", "|", "x0", "|", "f(x0)", "|", "f'(x_0)")
        print(f"{i}",f"|", f"{x_0}", "|", f"{f.subs(x, x_0)}", "|", f"{f_.subs(x, x_0)}" )
        x_0 = x_1.subs(x, x_0)
        if flag:
            x_1 = x_0 - f / f_
        else:
            x_1 = x_0 - f / f_.subs(x, x_tmp)
    return x_0
```

```
Iteration | x0 | f(x0) | f'(x_0)
1 | -1.2 | -0.744640000000000 | 19.2960000000000
2 | -1.83569494335108 | -54.9203732656342 | 182.200662066193
3 | -1.53426698670797 | -16.3997260290089 | 81.6165105090230
4 | -1.33333060843441 | -4.56367970416828 | 39.0118784106938
5 | -1.21634880981081 | -1.07711329232894 | 21.3907381855373
6 | -1.16599461992795 | -0.158992113021013 | 15.2081524786430
7 | -1.15554021968149 | -0.00622590232961251 | 14.0225762983991
-1.15509622834943
Iteration | x0 | f(x0) | f'(x_0)
1 | -0.7 | -1.18664000000000 | -11.7390000000000
2 | -0.801085271317829 | -0.0928889824478163 | -9.63255553474314
3 | -0.810728504841143 | -0.00137752146919157 | -9.34415172449000
-0.810875925549818
Iteration | x0 | f(x0) | f'(x_0)
1 | 0.5 | -3.50000000000000 | 13.1250000000000
2 | 0.766666666666667 | 0.618993251028808 | 17.4422345679012
3 | 0.731178481444807 | 0.00819813419430093 | 16.9754793411857
0.730695541668584
```

Рис. 9: Таблицы Модиф. Метода Ньютона

3.5.1 Вычисление погрешностей

$\delta_a = 8.133453463423e-06$, $d_a = 9.123324899948654e-08$, $\delta_b = 4.533647327654183e-06$, $d_b = 3.231920515504325e-08$, $\delta_c = 1.2345852309625743e-05$, $d_c = 9.423443223897342e-08$

3.6 Метод Секущих

```python
def secant(a, b, f, e, max) -> Float:
    """
    return: root of f(x) = 0
    """
    i = 0
    while abs(a - b) >= e and i < max:
        a = b - (b - a) * f.subs(x, b) / (f.subs(x, b) - f.subs(x, a))
        b = a - (a - b) * f.subs(x, a) / (f.subs(x, a) - f.subs(x, b))
        i = i + 1
        if i ==1:
            print("Iteration", "|", "a", "|", "b", "|", "f(a)", "|", "f(b)")
        print(f"{i}",f"|", f"{a}", "|", f"{b}", "|", f"{f.subs(x, a)}", "|", f"{f.subs(x,
            b)}" )
    return b
```

```
Iteration | a | b | f(a) | f(b)
1 | -1.03389830508475 | -1.81103565974114 | 0.958203681086520 | -50.5534648441412
2 | -1.04835436708508 | -1.06189226125360 | 0.913560018406660 | 0.856221746487638
3 | -1.26405111607743 | -1.11757721451876 | -2.25220876473050 | 0.448216915251542
4 | -1.14188896685031 | -1.15744724600909 | 0.174905402140478 | -0.0331710279193587
5 | -1.15496698376985 | -1.15509424052967 | 0.00179397692437178 | 0.0000167569400257861
6 | -1.15509544040021 | -1.15509543978420 | -8.60739213237594E-9 | 4.26325641456060E-14
[-1.5, -1]  -1.15509543978420
Iteration | a | b | f(a) | f(b)
1 | -0.891891891891892 | -0.833127091281473 | 0.639462859034710 | 0.199870968514794
2 | -0.806408268969507 | -0.811050730645855 | -0.0420310814295846 | 0.00163180788090145
3 | -0.810877228487159 | -0.810875960973896 | 0.0000118346513406919 | -3.41014860794076E-9
[-1, -0.5]  -0.810875960973896
Iteration | a | b | f(a) | f(b)
1 | 0.705882352941176 | 0.728514126717895 | -0.416813101601077 | -0.0369823642914797
2 | 0.730717677984707 | 0.730695429032144 | 0.000377214503201051 | -3.29706381752004E-7
[0.5, 1]  0.730695429032144
```

Рис. 10: Таблицы Модиф. Метода Ньютона

3.6.1 Вычисление погрешностей

$\delta_a = 9.611526341343468e-14$, $d_a = 1.1102230246251565e-15$, $\delta_b = 4.502873042796951e-08$, $d_b = 3.65127150736555e-10$, $\delta_c = 2.659111638749789e-06$, $d_c = 1.9430007713872044e-08$

3.7 Вторая функция

2 Методы и код методов остаются те же самыми, поэтому буду приводить таблицы значений. Для начала рассмотрим график функции:

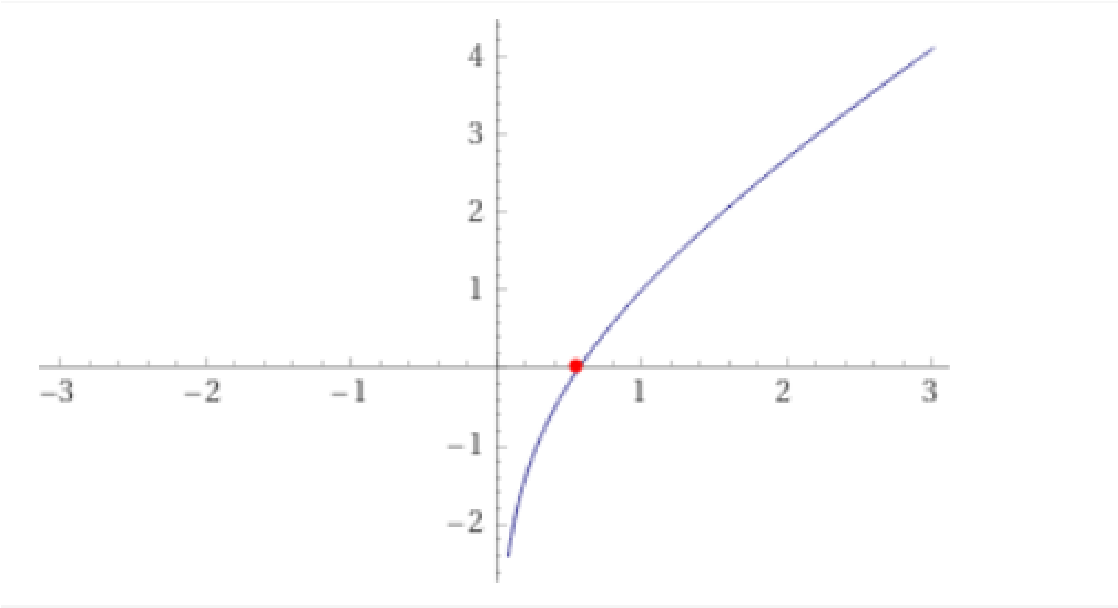

Рис. 11: График второй функции

Видим, что данная функция имеет один корень. Её отрезком локализации возьмем $[0.2, 1]$

3.7.1 Метод Деления Отрезка Пополам

```
Iteration | x_ | f(a) | f(b) | f((a+b)/2)
1 | 0.6 | -1.40943791243410 | 1.00000000000000 | 0.0891743762340093
2 | 0.4 | -1.40943791243410 | 0.0891743762340093 | -0.516290731874155
3 | 0.5 | -0.516290731874155 | 0.0891743762340093 | -0.193147180559945
4 | 0.55 | -0.193147180559945 | 0.0891743762340093 | -0.0478370007556204
5 | 0.575 | -0.0478370007556204 | 0.0891743762340093 | 0.0216147618152133
6 | 0.5625 | -0.0478370007556204 | 0.0216147618152133 | -0.0128641449035618
7 | 0.56875 | -0.0128641449035618 | 0.0216147618152133 | 0.00443569128302301
8 | 0.565625 | -0.0128641449035618 | 0.00443569128302301 | -0.00419896452794633
9 | 0.5671875 | -0.00419896452794633 | 0.00443569128302301 | 0.000122157911133125
10 | 0.56640625 | -0.00419896452794633 | 0.000122157911133125 | -0.00203745205898809
11 | 0.566796875 | -0.00203745205898809 | 0.000122157911133125 | -0.000957409589420366
12 | 0.5669921874999999 | -0.000957409589420366 | 0.000122157911133125 | -0.000417566508923772
13 | 0.5670898437499999 | -0.000417566508923772 | 0.000122157911133125 | -0.000147689471449053
Конечный результат = 0.5670898437499999
Абсолютная погрешность: 5.3156250000019334e-05
Относительная погрешность: 0.009372636178180694
```

Рис. 12: Метод Бисекции для 2-й функции

3.7.2 Метод простой итерации

Данный метод не подходит, потому что при выборе x_0 близких к x^* производная $\phi = -ln(x)$ в данной точке не будет удовлетворять условию сходимости данного метода. При выборе другого представления $x = \phi(x) = -e^x$ получаем, что при $x > 1$, производная больше 1 по модулю и не сходится при меньших($>$0). Вывод: данный метод не подходит для данной функции.

3.7.3 Метод Ньютона

```
F(x0)*F''(x0) = 10.0441422702882

x0 = 0.3
Iteration | x0 | f(x0) | f'(x_0)
1 | 0.3 | -0.903972804325936 | 4.33333333333333
2 | 0.508609108690601 | -0.167466408123344 | 2.96614646280023
3 | 0.565068359851531 | -0.00574020441530632 | 2.76969738716701
Конечный результат = 0.567140862227387
Абсолютная погрешность: 0.00000213777261259818
Относительная погрешность: 0.000376937141531885
```

Рис. 13: Таблица МН 2-й функции

3.7.4 Модифицированный Метод Ньютона

```
F(x0)*F''(x0) = 10.0441422702882

Iteration | x0 | f(x0) | f'(x_0)
1 | 0.3 | -0.903972804325936 | 4.33333333333333
2 | 0.508609108690601 | -0.167466408123344 | 2.96614646280023
3 | 0.565068359851531 | -0.00574020441530632 | 2.76969738716701
Конечный результат = 0.567140862227387
Абсолютная погрешность: 0.00000213777261259818
Относительная погрешность: 0.000376937141531885
```

Рис. 14: Таблица модеф-го МН 2-й ф-ии

3.7.5 Метод Секущих

```
Iteration | a | b | f(a) | f(b)
1 | 0.2 | 1.0 | -1.40943791243410 | 1.00000000000000
2 | 0.667972353273129 | 0.548590982121758 | 0.264463859655483 | -0.0518111564763825
3 | 0.568147654592244 | 0.567153887464539 | 0.00277371581677832 | 0.0000292818491143265
Конечный результат = 0.567143290409820
Абсолютная погрешность: -2.90409819636572E-7
Относительная погрешность: -0.0000512057487505924
```

Рис. 15: Таблица метода секущих