УО «Белорусский государственный университет информатики и радиоэлектроники»

Отчет по лабораторной работе №2

по предмету «Численные методы»

Вариант 14

Выполнил:

Наривончик А.М.

Гр. 351004

Проверил:

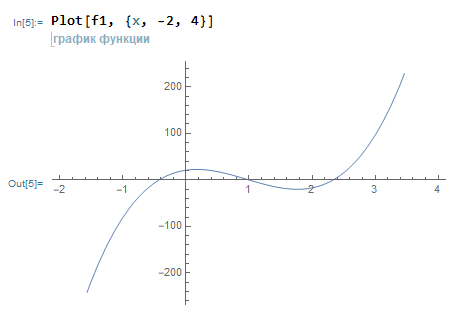
Степанова Т. С.

Минск 2024

**Численное решение нелинейных уравнений**

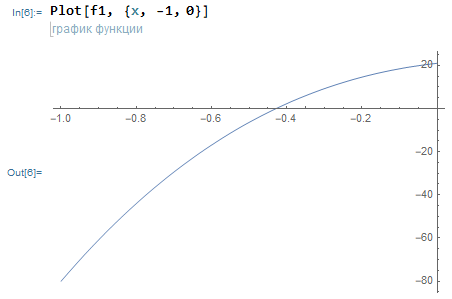
1. Отделите графически корни алгебраического уравнения f (x) = 0 с помощью функции **Plot.** Найдите один из них (нецелый) с точностью ε = 10−3 методом хорд. Укажите потребовавшееся число итераций. Проиллюстрируйте графически нахождение первых двух приближений (постройте график функции и хорды).

****



**Отделение корней:**

Первый корень находится на отрезке [-0.6; -0.2].



f(x)’’ на отрезке [a, b] < 0, значит, для сходимости метода хорд, в качестве первого приближения возьмём x0 = b = -0.2.

Зададим все входные данные, а также функции для задания хорд и для вычисления следующего приближения и погрешности:

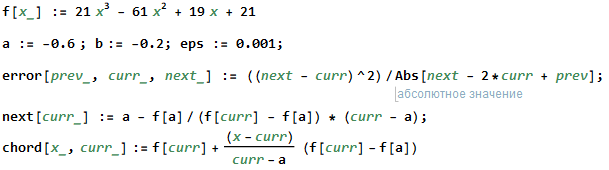
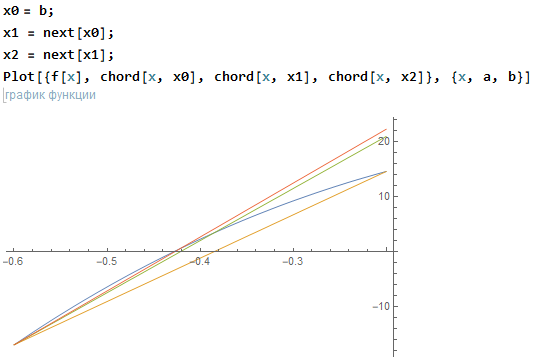
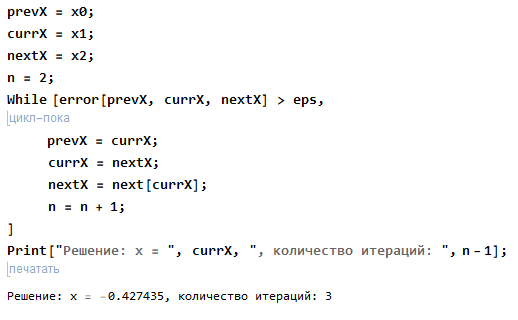


График функции и первые два шага метода хорд:



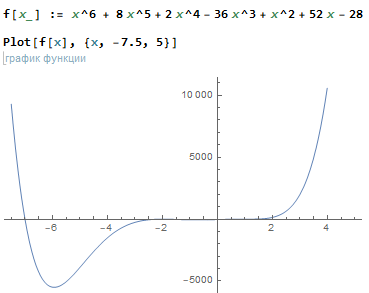


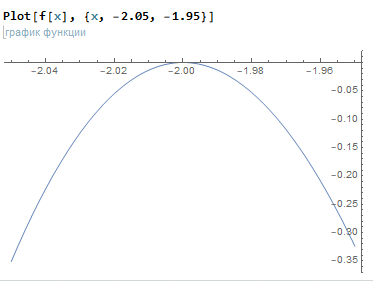
Округленное значение корня: х = -0.427.

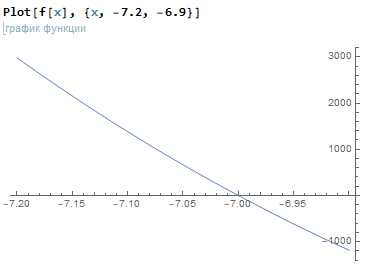
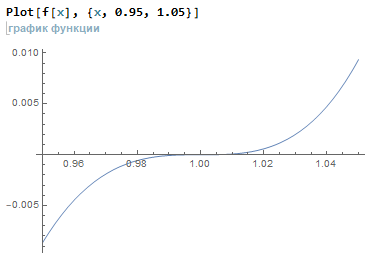
1. Отделите графически и найдите с помощью функций **Solve, NSolve, Roots, FindRoot** корни алгебраического уравнения f (x) = 0. Разложите многочлен f (x) на множители, используя функцию **Factor.**



График функции выглядит следующим образом:

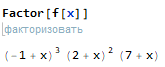


Отделим корни:

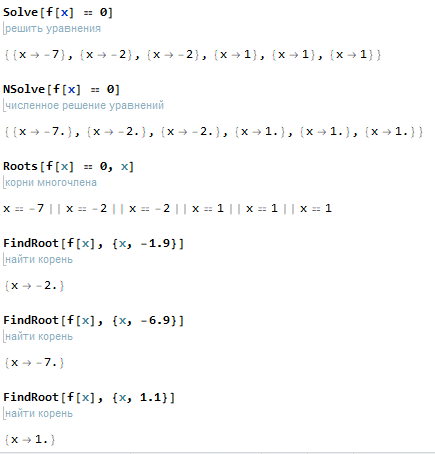


Получаем: X1 – [-7.05; -6.95], X2 - [-2.02; -1.98], X3 – [0,98; 1.02].

Факторизация многочлена дает результат:



Поиск корней встроенными функциями:

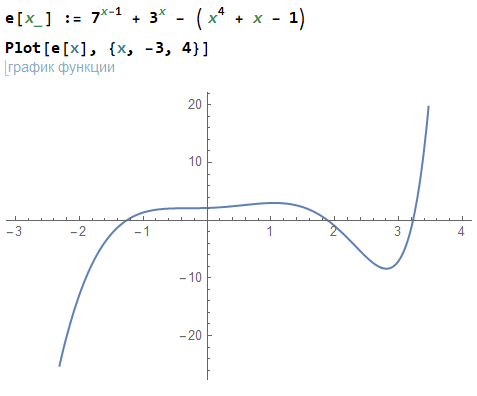


Видим, что все методы дают решение -7, -2 и 1, причем последние два корня – корни кратности 2 и 3 соответственно.

1. Отделите графически корни трансцендентного уравнения с помощью функции **Plot**. Найдите один из них с точностью ε = 10−3: а) методом Ньютона; б) методом секущих. Укажите потребовавшееся число итераций.



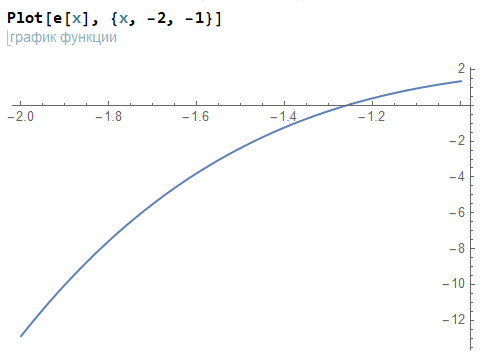
Перенесём правую часть уравнения в левую и будем решать эквивалентное уравнение.



По графику видно, что уравнение имеет 3 корня: X1 – (-2, -1), X2 – (1.5, 2),

X3 – (3, 3.5).

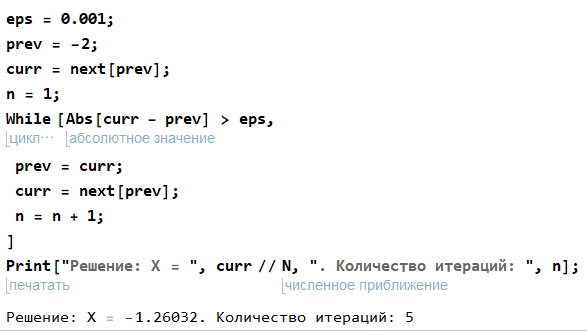
Методом Ньютона вычислим первый:



По участку графика видно, что первая и вторая производная функции на интервале (-2, -1) сохраняют знак, причем f’’ < 0, поэтому в качестве первого приближения к корню стоит взять точку x0 = -2, так как f(-2) \* f’’(-2) > 0, и скорость сходимости метода в этом случае будет квадратичной.

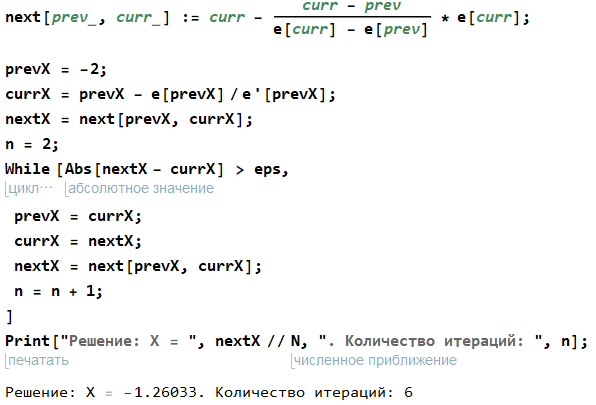
Зададим функцию вычисления следующего приближения.





Округленное значение: Х = -1.260.

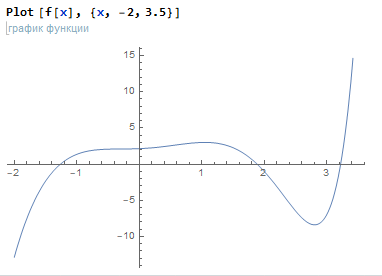
Вычислим этот корень методом секущих:



Округлённое значение: Х = -1.260.

1. Приведите уравнение к виду, пригодному для итераций. Найдите его корни методом простых итераций с точностью ε = 10−3. Укажите потребовавшееся число итераций.





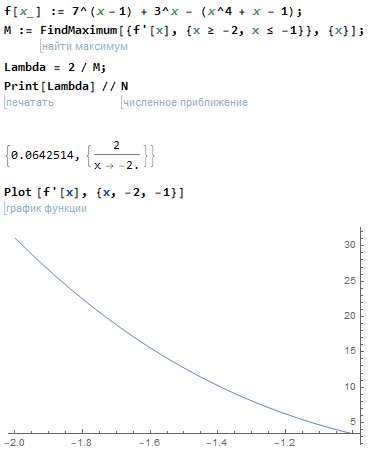
По графику видно, что уравнение имеет 3 корня: X1 – (-2, -1), X2 – (1.5, 2),

X3 – (3, 3.5).

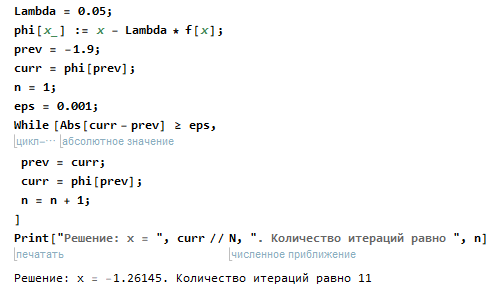
Приведём функцию к виду чтобы для любого х с отрезка [-2, -1].



 отрезке [-2, -1].

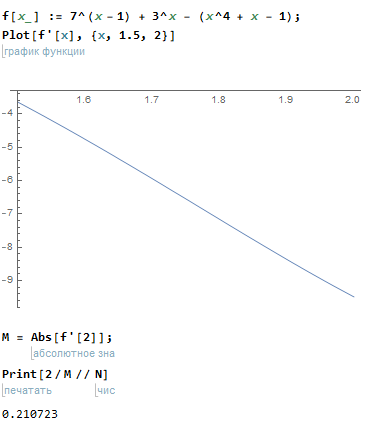


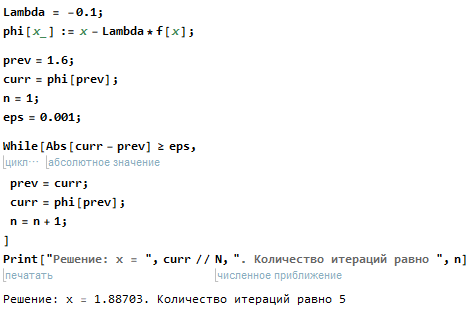
Учитывая знак производной, будем считать лямбда равной 0,05, а первое приближение – любое число на отрезке [-2, -1] – число -1.9.



После округления X = -1.261.

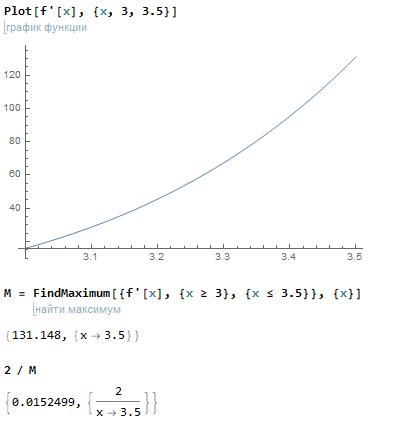
Аналогично найдем второй корень:

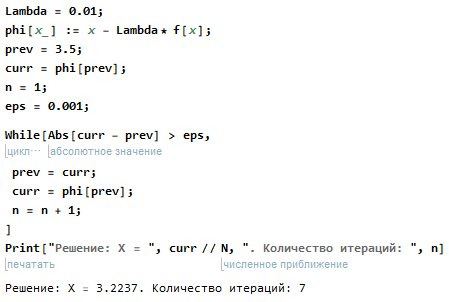




Округлённое значение: 1.887;

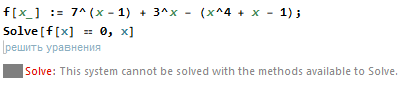
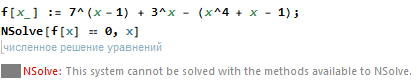
Третий корень:



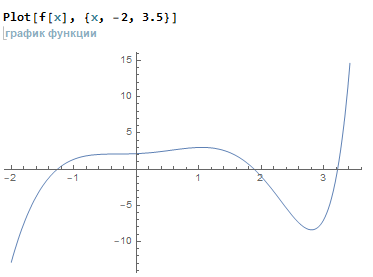


Округлённое значение равно 3.224.

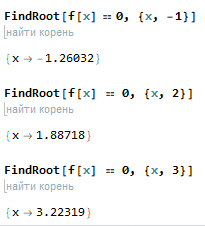
1. Решите уравнение (3.14) с помощью функций **Solve, NSolve, FindRoot**.

При попытке численно решить данное уравнение с помощью встроенных функций **Solve** и **NSolve** программа выдает сообщение о том, что это уравнение не может быть решено методами, доступными **Solve** и **NSolve**.



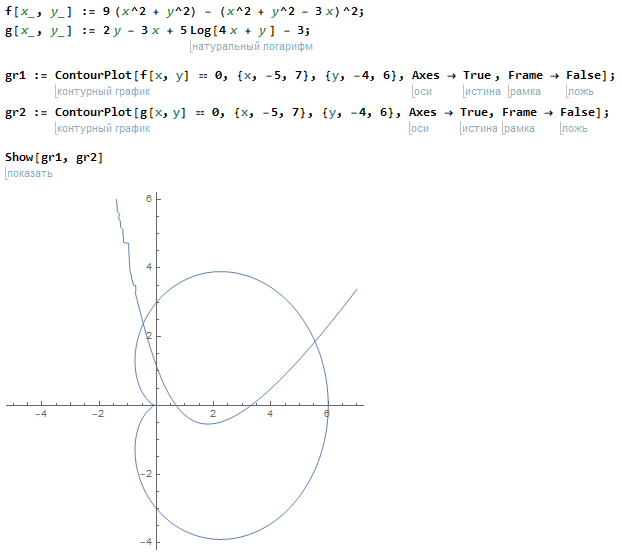
На графике видно, что уравнение имеет 3 корня, попробуем найти их с помощью **FindRoot**:



1. Дана система двух нелинейных уравнений f(x, y) = 0, g(x, y) = 0. Используя средства пакета Mathematica, изобразите на одном чертеже кривые f (x, y) = 0 и g(x, y) = 0 , и решите данную систему.



Решением данной системы являются точки пересечения кривых f (x, y) = 0 и g(x, y) = 0. Построим их с помощью функции **ContourPlot**, а затем объединим в одном графическом окне функцией **Show**.

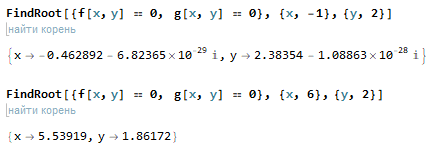


Система уравнений имеет 2 решения, причем в следующих промежутках:

Х1 – (-1,0), Y1 – (2,3),

X2 – (5, 6), Y2 – (1.5, 2.5)

Найдем корни с помощью функции FindRoot:



Действительного решения на промежутке (-1, 0) не оказалось, поэтому действительный корень систем только один.