Лабораторная №3

Задание №1

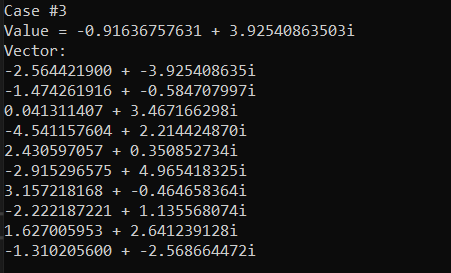
Для поиска максимального собственного значения и вектора, соответствующего этому значению, используем степенной метод.

Критерий для остановки 1-2 случаев: собственное значение на k-ой и k+1 итерациях различаются не больше, чем на eps.

В 3 случаем критерий остановки: модуль разницы собственных значений на k-ой и k+1 итерациях не больше, чем eps.

Для первой матрицы: выполнился 3 случай (с комплексными значениями)

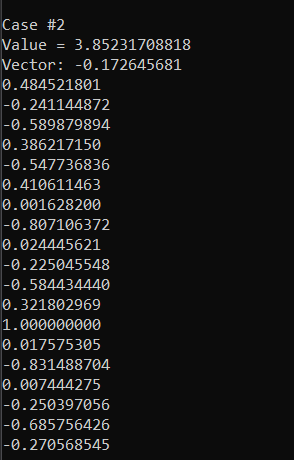
Начальное приближение: ya = {1, -1, 1, 1, 1, -1,2, -2,1, 2}, eps =

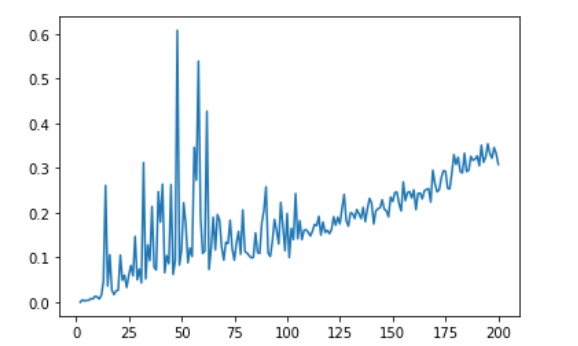


Для первой матрицы: выполнился 2 случай (максимальные значения равны по модулю, различаются по знаку)

Начальное приближение:

yb = {1, -1, 1, 1, 1, -1,2, -2,1, 2, 1, -1, 1, 1, 1, -1,2, -2,1, 2}, eps =



Запустим степенной метод на матрицах размером от 2 до 200, заполненных случайными числами, и построим график зависимости времени работы от размера матрицы. 

Из графика видно, что в целом влияние размера матрицы есть, но на некоторых небольших матрицах (30\*30, например), алгоритм работает гораздо дольше, чем на больших. Это объясняется тем, что время работы алгоритма зависит от того, насколько сильно максимальное собственное значение отличается от всех остальных.

Задание №2

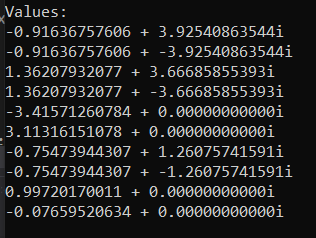
Для QR алгоритма сначала приведем матрицу к форме Хессенберга (использую метод простых итераций). Далее мы приводим матрицу к виду, где под главной диагональю находится хотя бы (n – 1) / 2 нулей. В теории сказано, что достаточно привести к форме Хессенберга только 1 раз (перед запуском самого алгоритма), но путём попыток улучшить точность значений выяснилось, что точность будет выше, если приводить к Хессенбергу на каждой итерации приведения к нужной форме. После этого мы ищем собственные значения.

Критерий остановки:

a(k) - значения полученные на k-ой итерации;

a(k + 1) - значения полученные на k + 1-ой итерации;

Тогда алгоритм остановится, если .

Собственные значений для 1ой матрицы: 

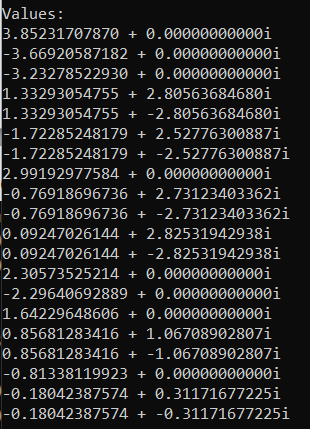
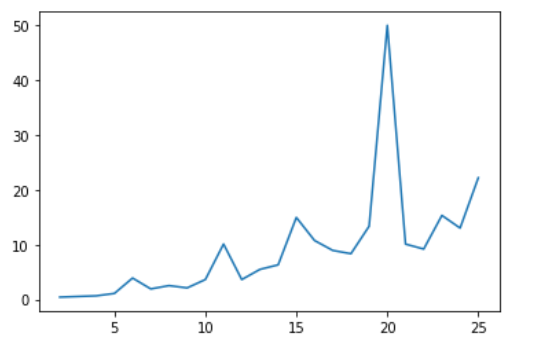
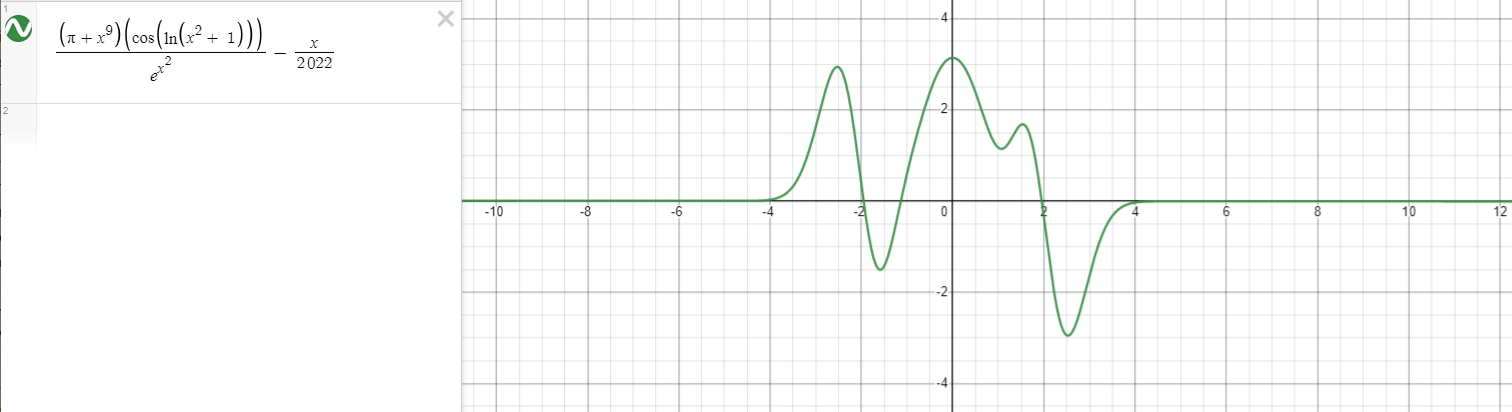
Собственные значения для 2ой матрицы: 

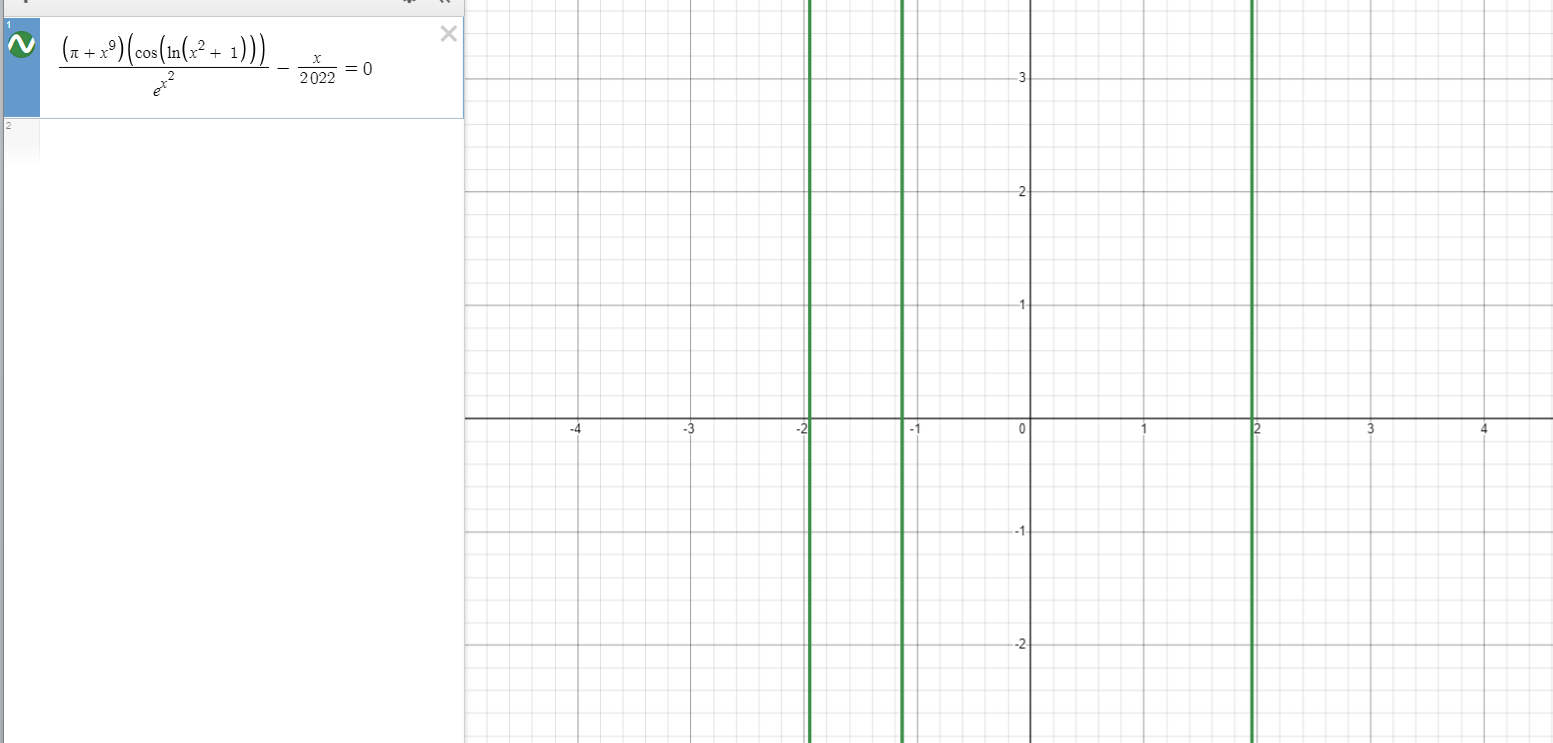
График зависимости времени работы алгоритма от размера матрицы: 

По графику можно сказать, что сильной зависимости размер/время нет. Очевидно, что для матриц с большим размером будут дольше работать вспомогательные операции (умножение матриц, поиск нулей, приведение к Хессенбергу). Сходимость же самого алгоритма зависит от того, насколько сильно собственные значения различаются между собой.

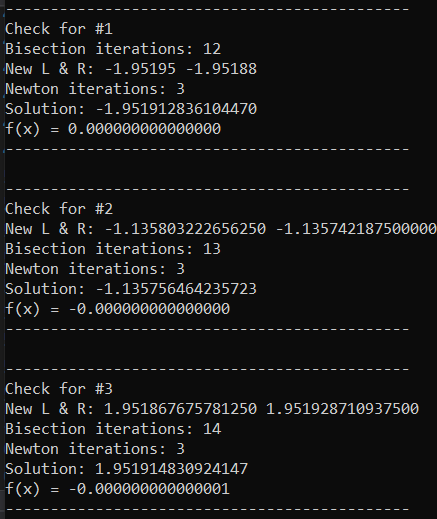
Задание №3

С помощью Desmos построим график нашей функции. 

Так же посмотрим на график f(x) = 0. Выберем отрезки отделенности [-2,-1.7], [-1.5 , -1], [1.6, 2.6]. Из графика видно, что на этих отрезках по одному корню.



С помощью бисекции сократим наши отрезки до длины . Т.к. производная нашей функции сложная, будем решать используя модификацию метода Ньютона.



Как видим, наш алгоритм правильно находит корни на заданных промежутках. Несмотря на то, что использовалась модификация алгоритма Ньютона, корни нашлись за маленькое число итераций.