

Факультет ПИиКТ

Тестирование программного обеспечения

Лабораторная работа 1

Вариант 27373244

Выполнил: Давтян Д. А.

Группа: P33022

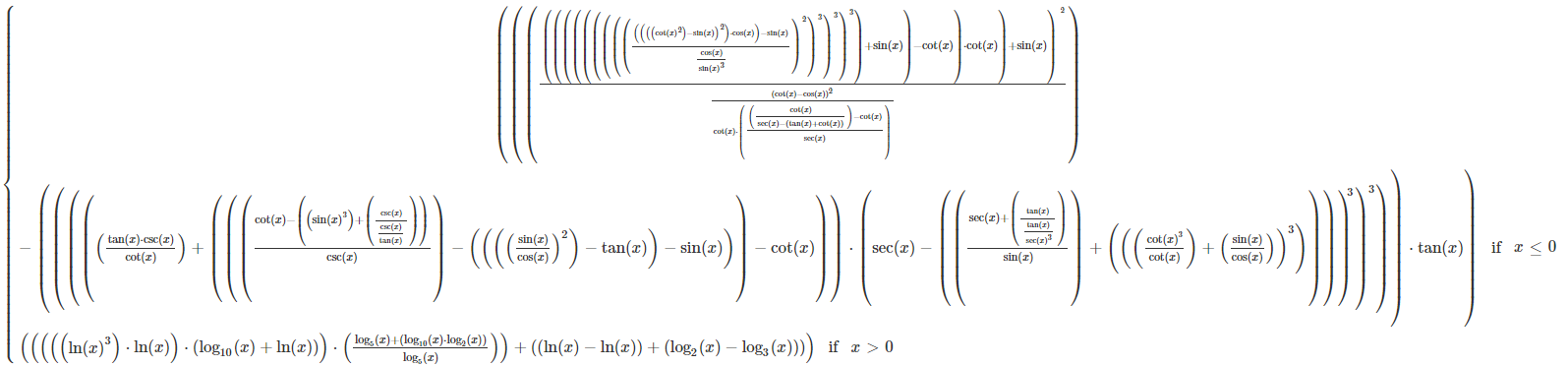
Преподаватель: Исаев И. В.

Санкт-Петербург

2021 г.

Задание

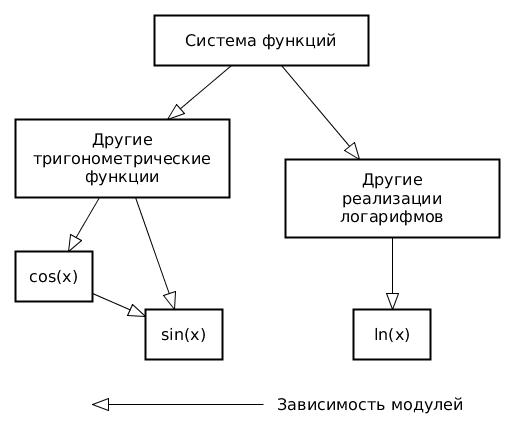
Провести интеграционное тестирование программы, осуществляющей вычисление системы функций (в соответствии с вариантом).



**Текстом:**x <= 0 : ((((((((((((((((((cot(x) ^ 2) - sin(x)) ^ 2) \* cos(x)) - sin(x)) / (cos(x) / (sin(x) ^ 3))) ^ 2) ^ 3) ^ 3) ^ 3) + sin(x)) - cot(x)) \* cot(x)) + sin(x)) ^ 2) / (((cot(x) - cos(x)) ^ 2) / (cot(x) \* (((cot(x) / (sec(x) - (tan(x) + cot(x)))) - cot(x)) / sec(x))))) - ((((((tan(x) \* csc(x)) / cot(x)) + ((((cot(x) - ((sin(x) ^ 3) + (csc(x) / (csc(x) / tan(x))))) / csc(x)) - ((((sin(x) / cos(x)) ^ 2) - tan(x)) - sin(x))) - cot(x))) \* (sec(x) - (((sec(x) + (tan(x) / (tan(x) / (sec(x) ^ 3)))) / sin(x)) + ((((cot(x) ^ 3) / cot(x)) + (sin(x) / cos(x))) ^ 3)))) ^ 3) ^ 3)) \* tan(x))

x > 0 : (((((ln(x) ^ 3) \* ln(x)) \* (log\_10(x) + ln(x))) \* ((log\_5(x) + (log\_10(x) \* log\_2(x))) / log\_5(x))) + ((ln(x) - ln(x)) + (log\_2(x) - log\_3(x))))

**Правила выполнения работы:**

1. Все составляющие систему функции (как тригонометрические, так и логарифмические) должны быть выражены через базовые (тригонометрическая зависит от варианта; логарифмическая - натуральный логарифм).
2. Структура приложения, тестируемого в рамках лабораторной работы, должна выглядеть следующим образом (пример приведён для базовой тригонометрической функции sin(x)):  
   
3. Обе "базовые" функции (в примере выше - sin(x) и ln(x)) должны быть реализованы при помощи разложения в ряд с задаваемой погрешностью. Использовать тригонометрические / логарифмические преобразования для упрощения функций ЗАПРЕЩЕНО.
4. Для КАЖДОГО модуля должны быть реализованы табличные заглушки. При этом, необходимо найти область допустимых значений функций, и, при необходимости, определить взаимозависимые точки в модулях.
5. Разработанное приложение должно позволять выводить значения, выдаваемое любым модулем системы, в сsv файл вида «X, Результаты модуля (X)», позволяющее произвольно менять шаг наращивания Х. Разделитель в файле csv можно использовать произвольный.

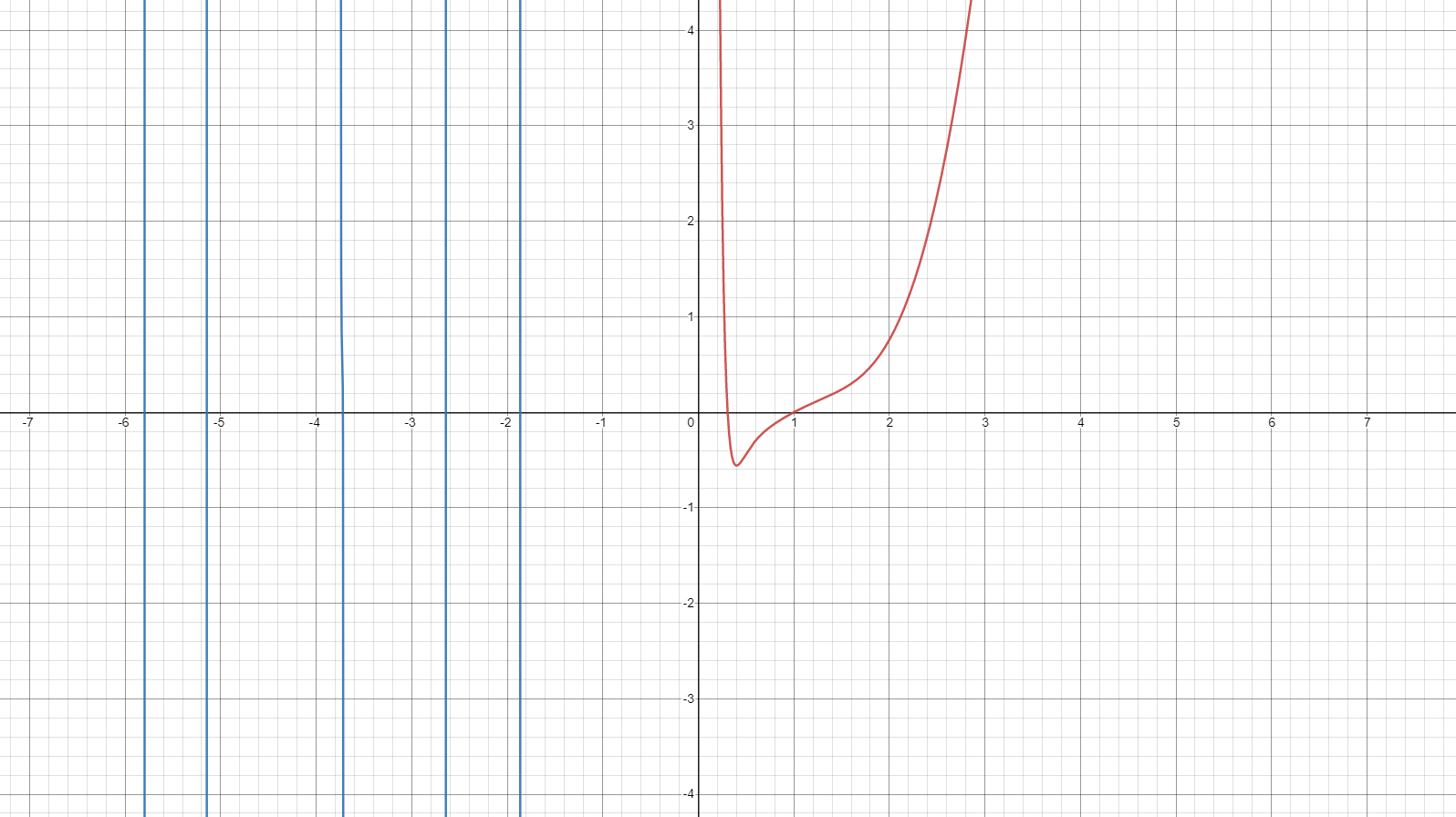
**Порядок выполнения работы:**

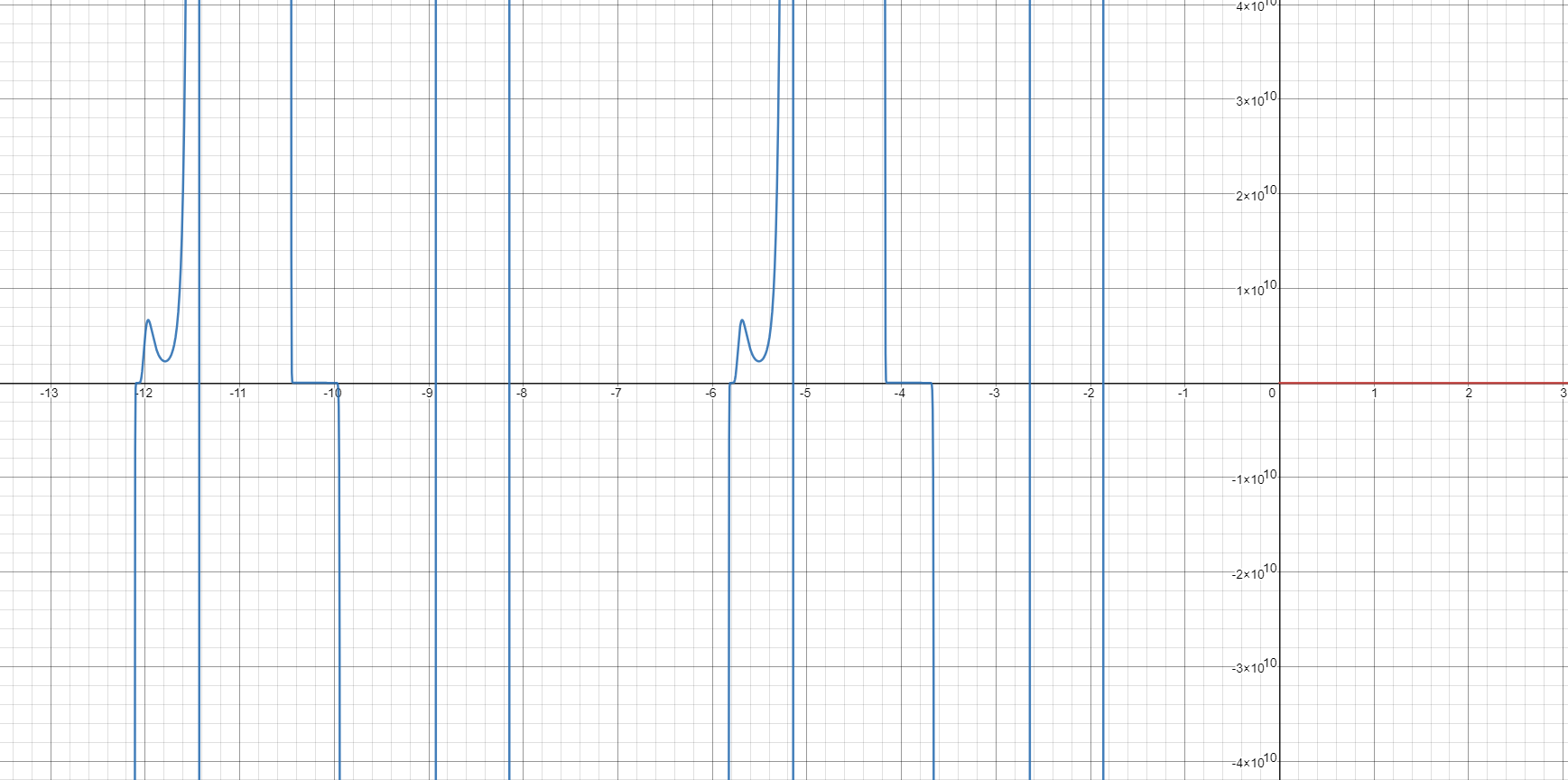
1. Разработать приложение, руководствуясь приведёнными выше правилами.
2. С помощью JUnit 5 разработать тестовое покрытие системы функций, проведя анализ эквивалентности и учитывая особенности системы функций. Для анализа особенностей системы функций и составляющих ее частей можно использовать сайт <https://www.wolframalpha.com/>.
3. Собрать приложение, состоящее из заглушек. Провести интеграцию приложения по 1 модулю, с обоснованием стратегии интеграции, проведением интеграционных тестов и контролем тестового покрытия системы функций.

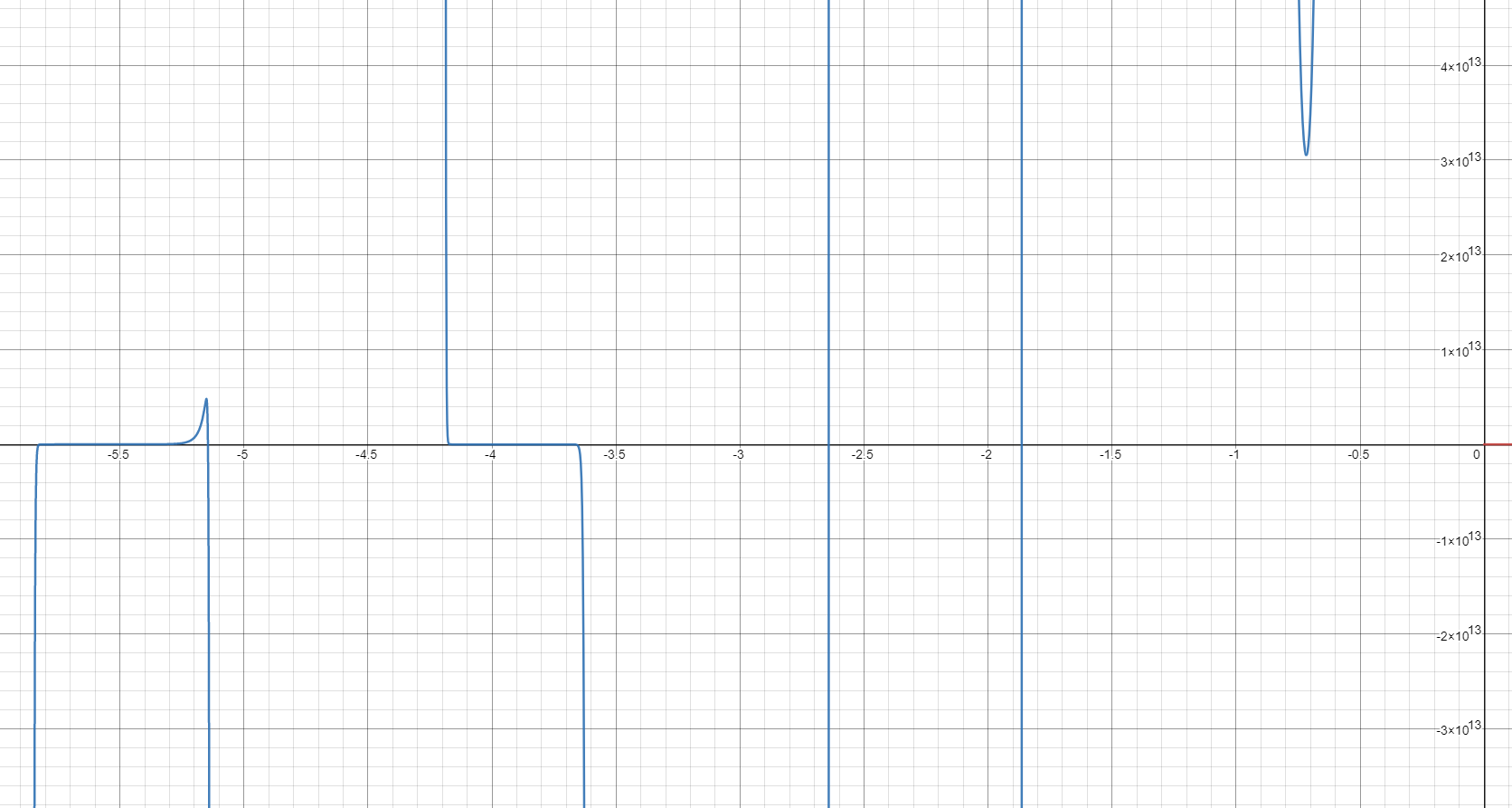
**Отчёт по работе должен содержать:**

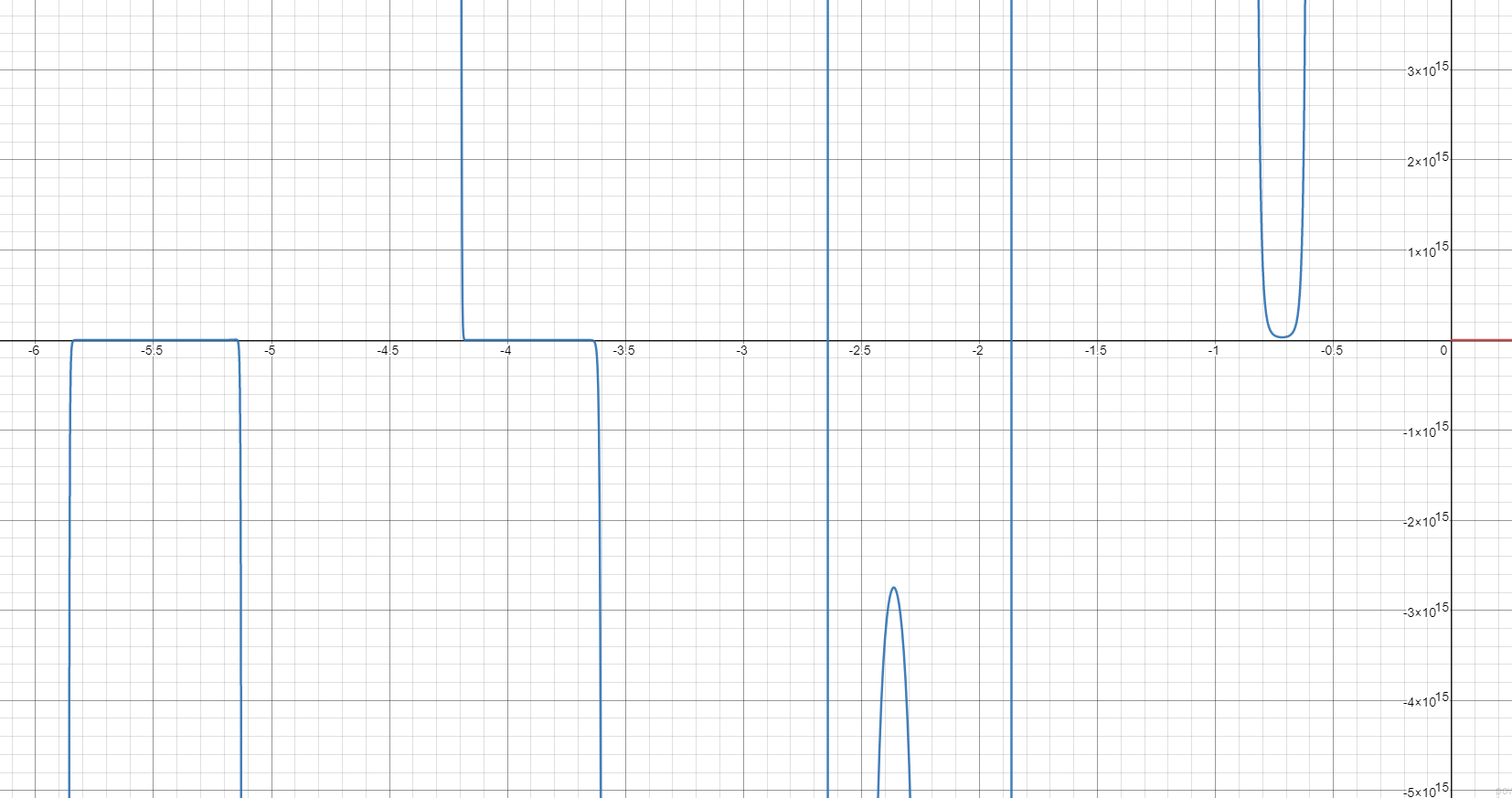
1. Текст задания, систему функций.
2. UML-диаграмму классов разработанного приложения.
3. Описание тестового покрытия с обоснованием его выбора.
4. Графики, построенные csv-выгрузкам, полученным в процессе интеграции приложения.
5. Выводы по работе.

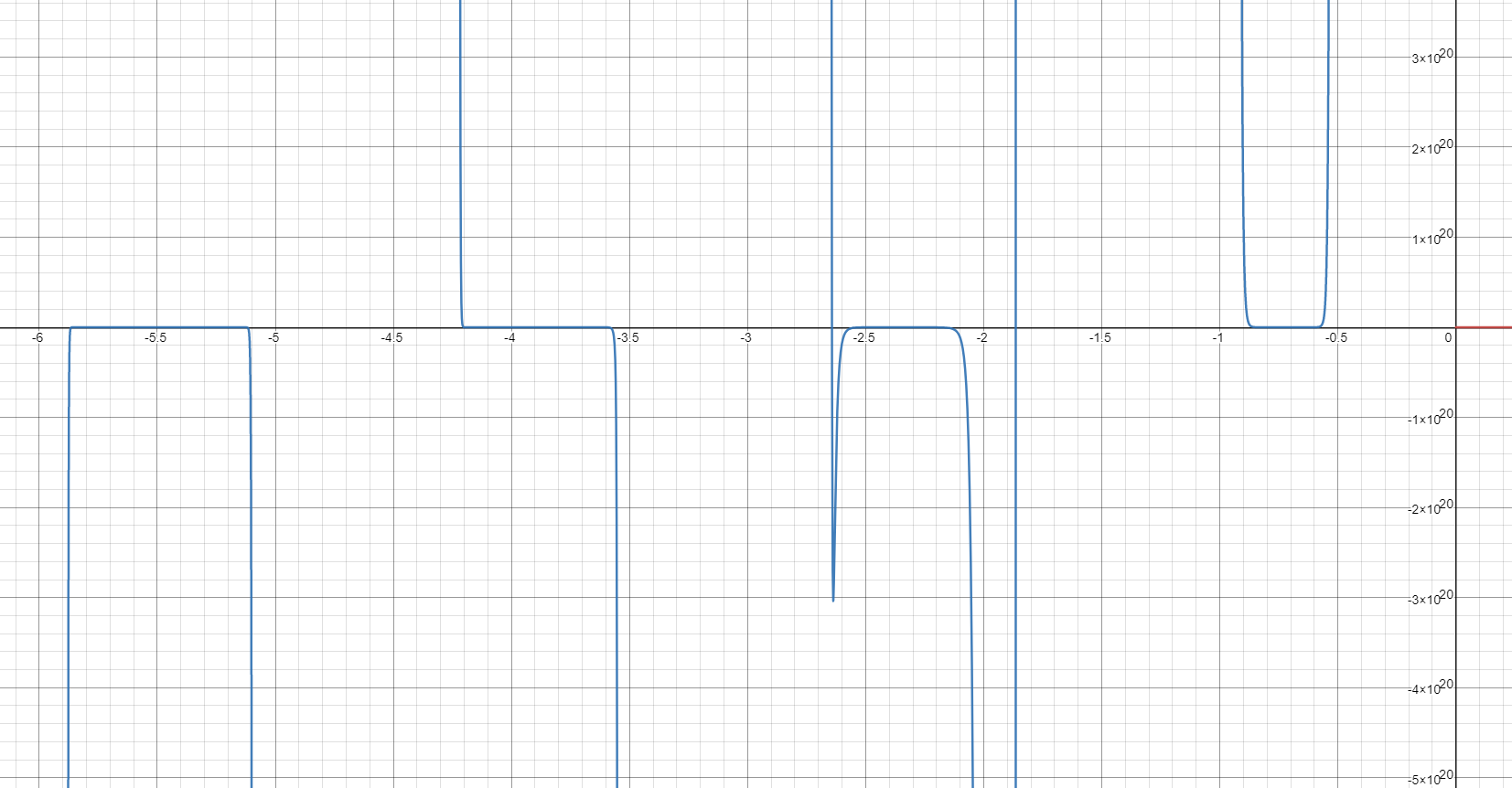
Графики

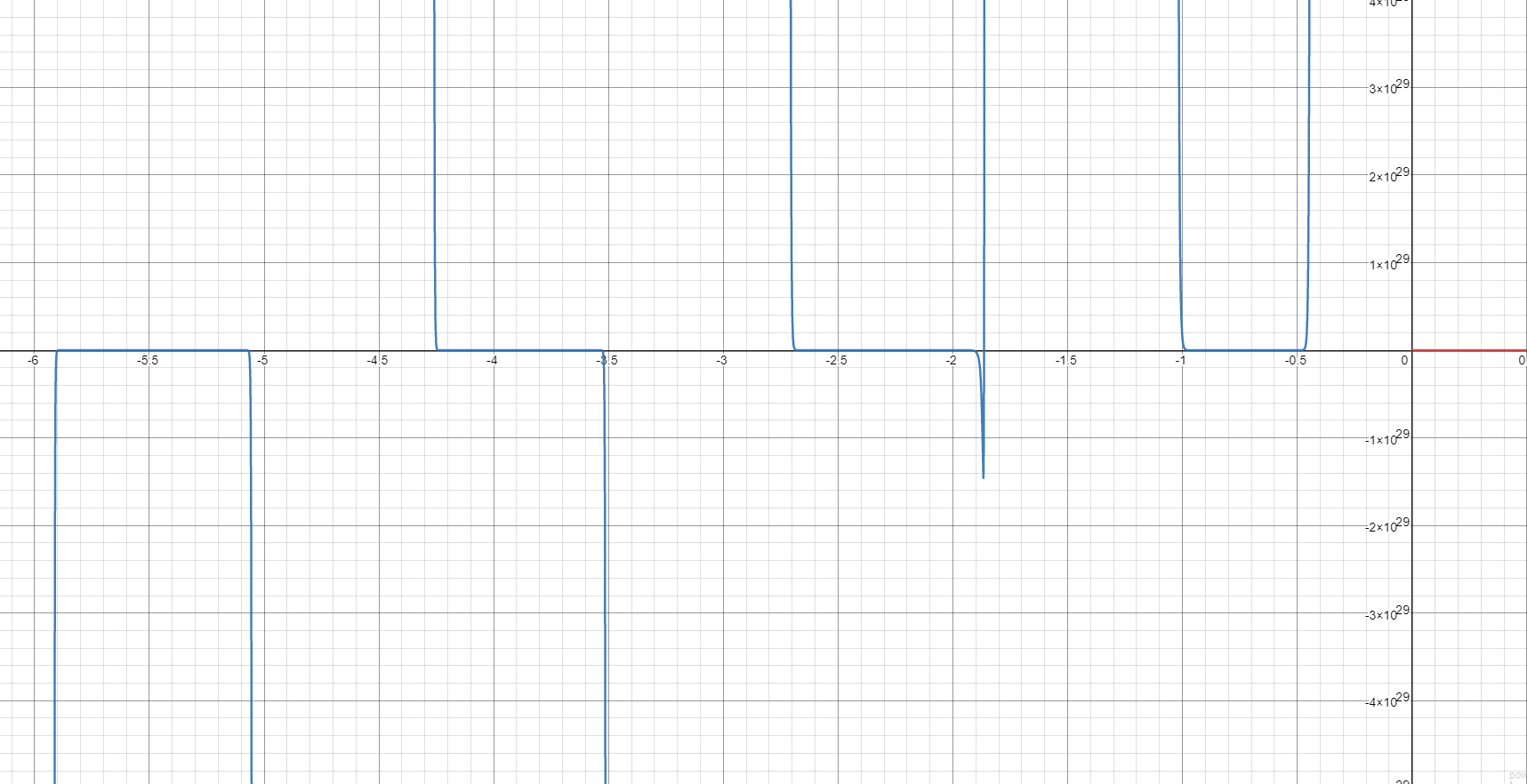
  
График при стандартном масштабе

  
График при масштабе Y в пределах

  
График при масштабе Y в пределах

  
График при масштабе Y в пределах

  
График при масштабе Y в пределах

  
График при масштабе Y в пределах

Анализ

Для

Функция имеет асимптоты на   
Функция имеет период   
Функция имеет экстремумы на   
Функция пересекатеся с на

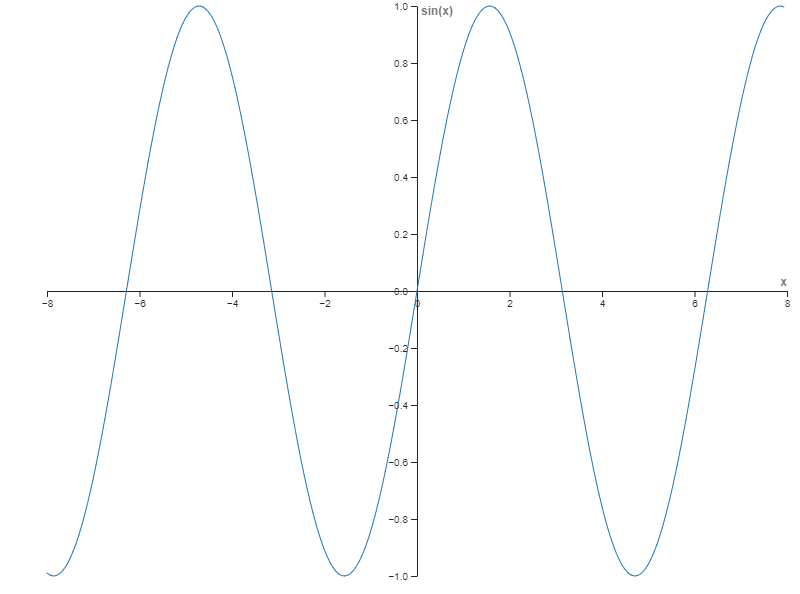
Для

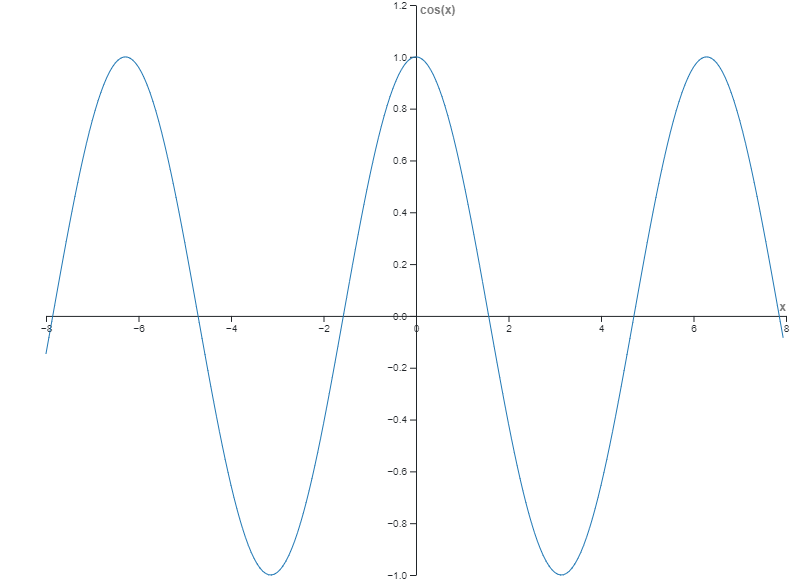
Функция имеет асимптоту на   
Функция имеет экстремум на   
Функция не определена на   
Функция пересекается с на

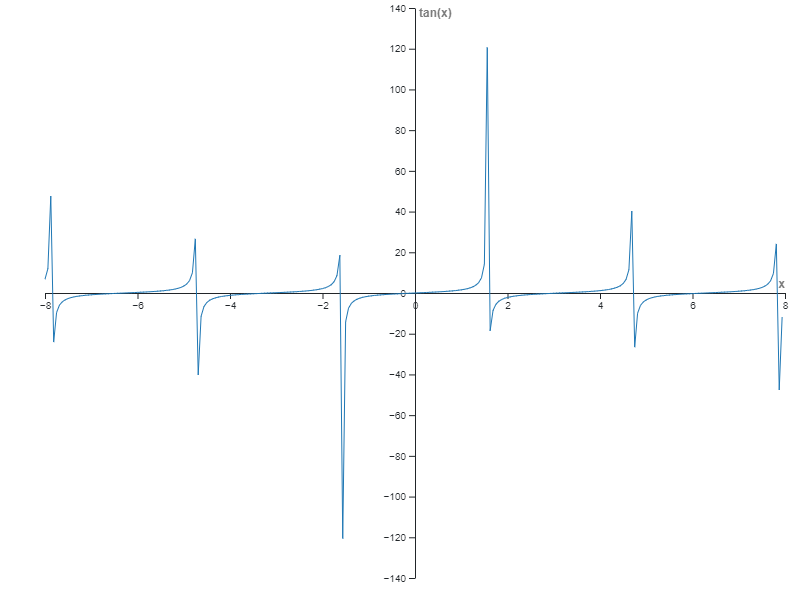
Код

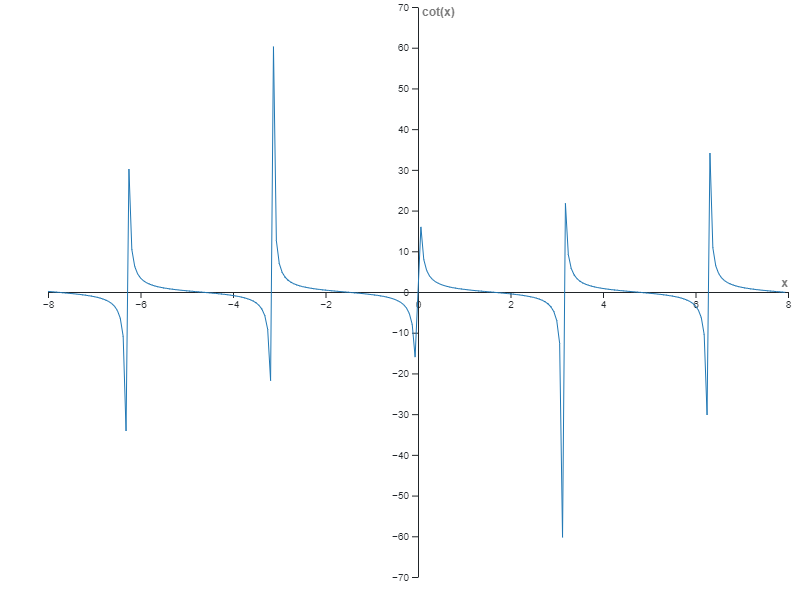
<https://github.com/david-d25/testing-labwork2>

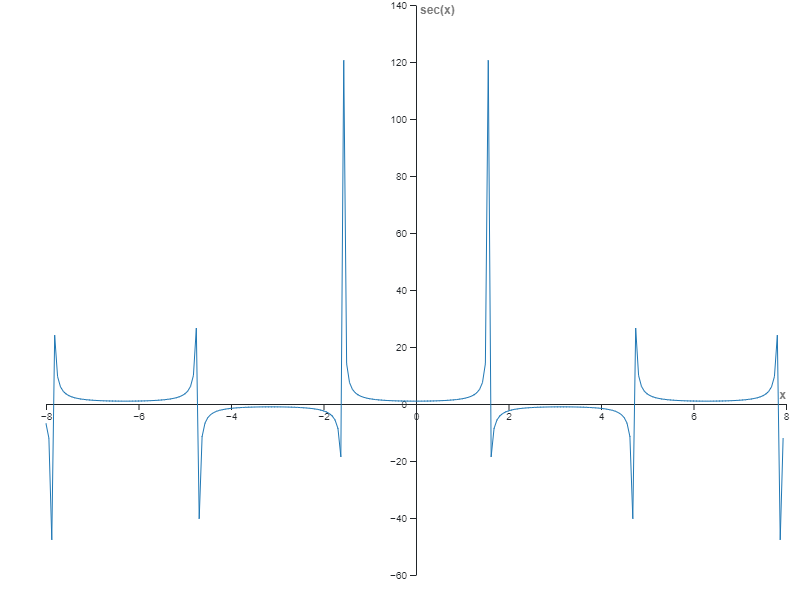
Графики, построенные по CSV

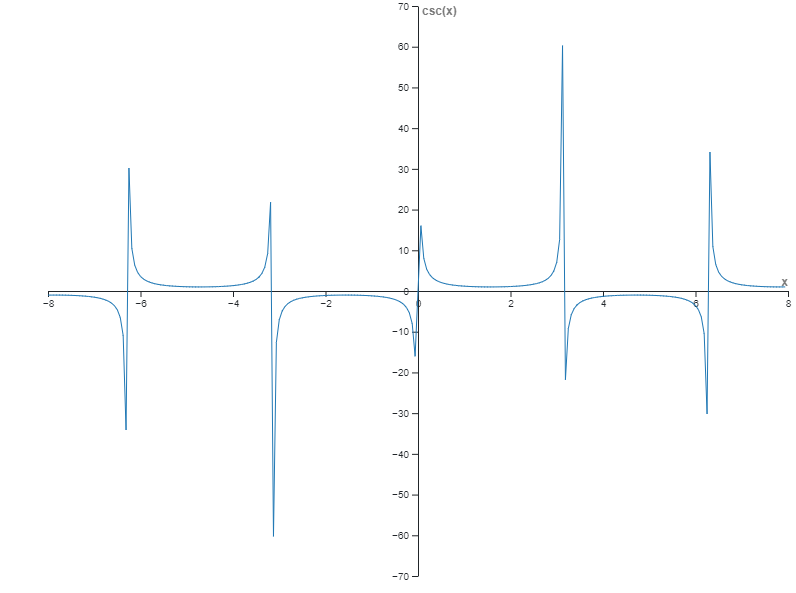


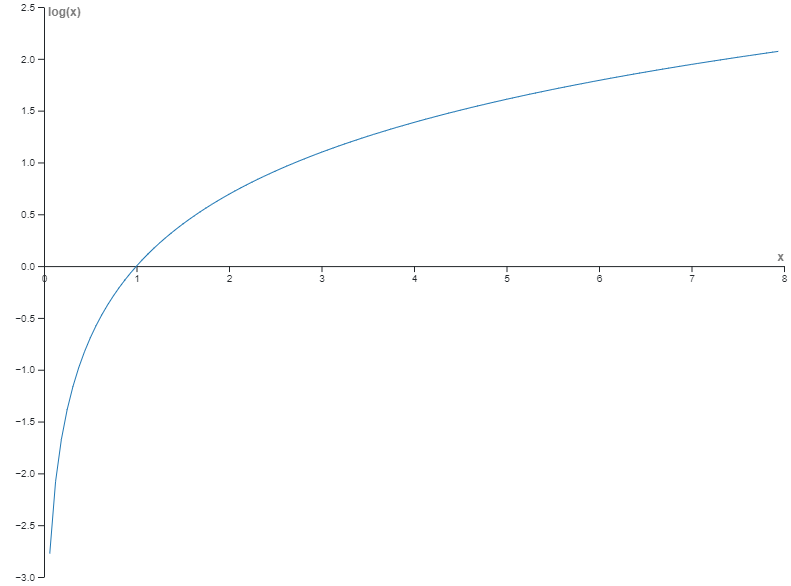


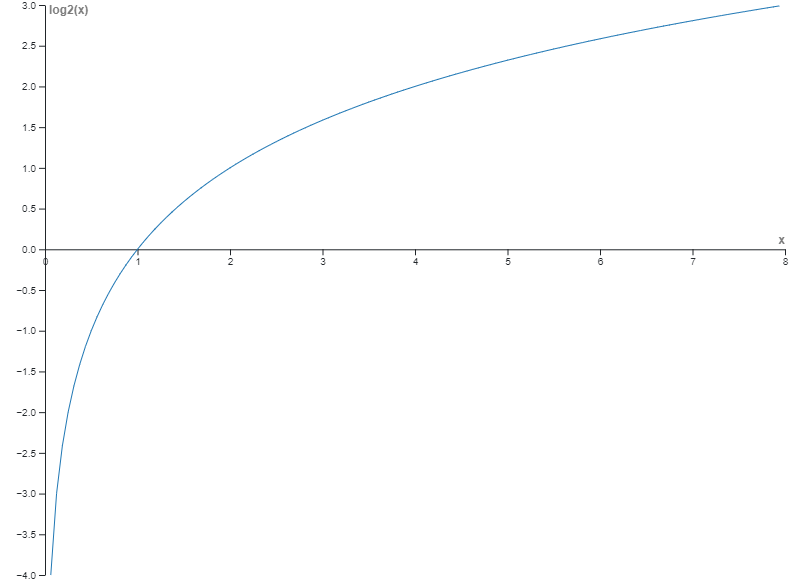


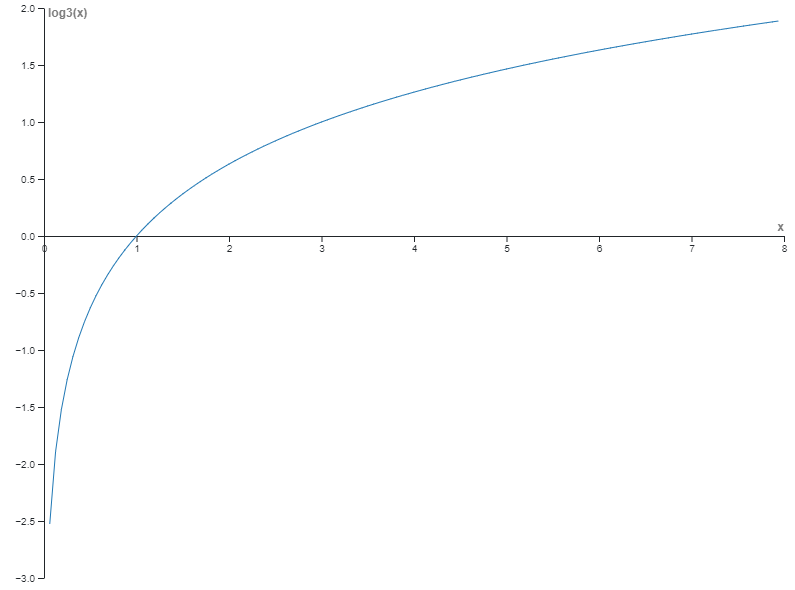


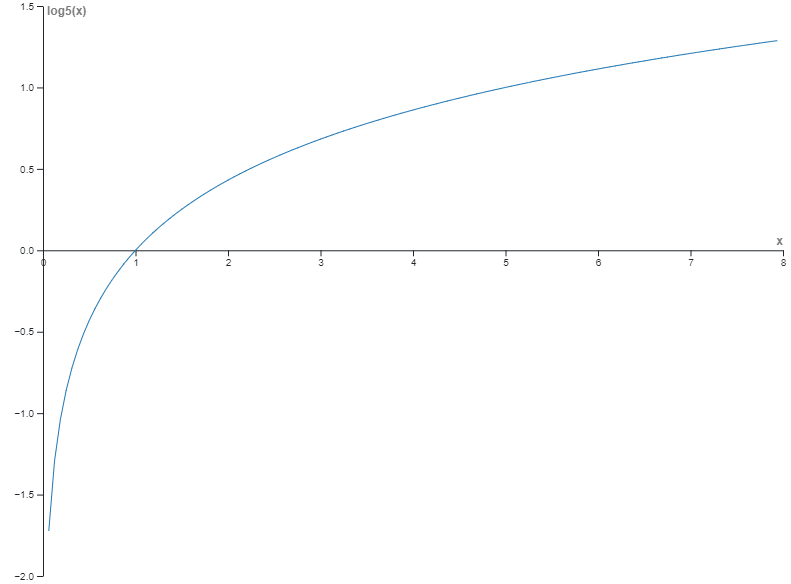


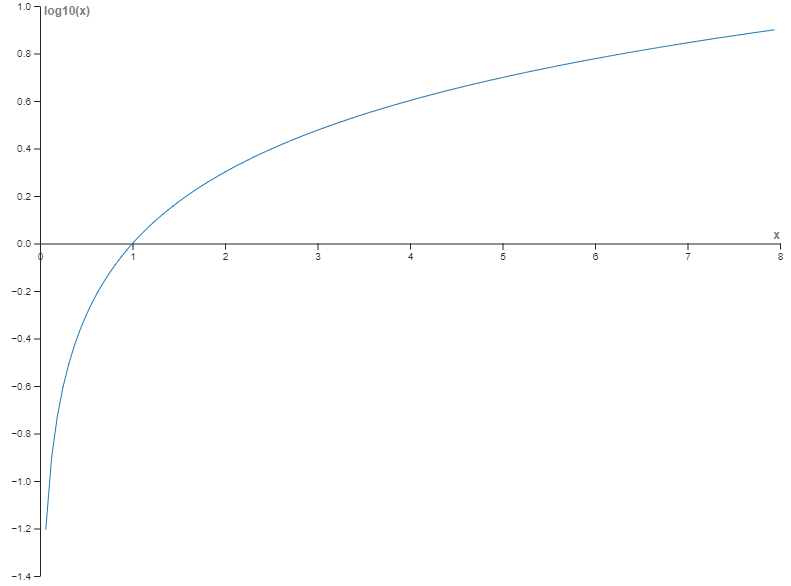


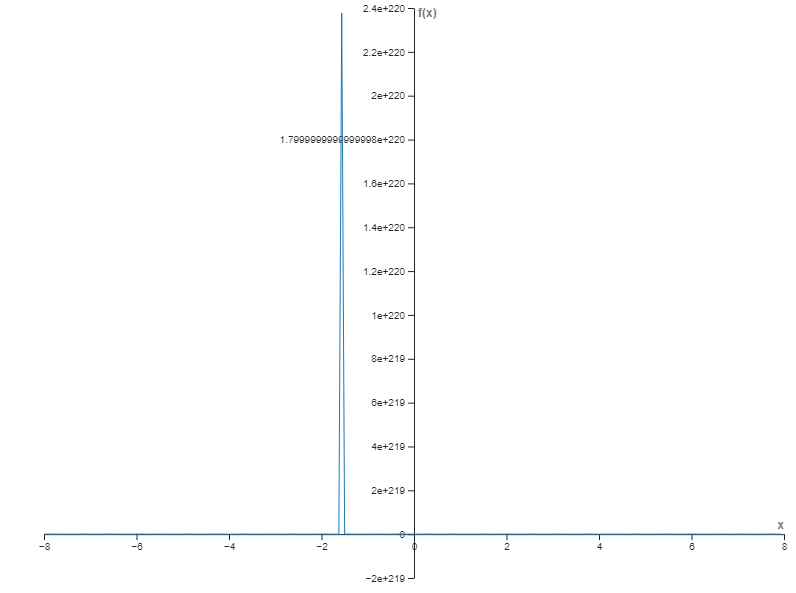












Вывод

При выполнении этой лабораторной работы было изучено интеграционное тестирование и Mockito. Для этой лабораторной работы использовалась стратегия «сверху вниз» с тремя этапами. На первом этапе проверялась система функций, а тригонометрические и логарифмические функции были заглушками. На втором этапе заглушами были только базовые функции (синус и натуральный логарифм). На третьем этапе тестирование проводилось без заглушек.

Данную по варианту функцию оказалось очень сложно тестировать из-за её очень большого масштаба: различные граничные точки находились на 20-30 порядках, это увеличи