# Техническое задание (ТЗ):

1. Описание программы:
   * Программа предназначена для нахождения точек пересечения кривых и приближенного решения уравнений.
   * Основная программа реализует различные методы поиска корней и вычисления интегралов.
   * Входные данные представляют собой интервал и функцию, заданную пользователем.
2. Функциональные требования:
   * Программа должна поддерживать следующие методы поиска корней: a. Метод бисекции (Bisection method). b. Метод линейного поиска (Line search method). c. Метод хорд (Chord method). d. Метод касательных (Tangent method). e. Метод комбинированного поиска (Combined method).
   * Программа должна поддерживать следующие методы вычисления интегралов: f. Метод прямоугольников (Rectangles method). g. Метод тrapezoidal (Integral Trapezoidal method). h. Метод Симпсона (Integral Simpson method). i. Метод Монте-Карло (Integral Monte Carlo method).
   * Программа должна позволять пользователю выбрать метод и вводить необходимые параметры (интервал, точность и другие).
3. Дополнительные требования:
   * Основная программа должна поддерживать опции командной строки, при задании которых: a. Печатаются абсциссы точек пересечения кривых/корни уравнения b. Печатается число итераций, потребовавшихся на приближённое решение уравнений при поиске точек пересечения.
   * Программа должна поддерживать ключ командной строки "--help", выводящий на печать все допустимые ключи командной строки.
4. Прочие требования:
   * Программа должна быть написана на языке программирования C++.
   * Для реализации методов поиска корней и вычисления интегралов могут использоваться соответствующие классы и функции.
   * Код должен быть хорошо комментирован и иметь понятную структуру.
   * Результаты вычислений должны выводиться на экран с соответствующими пояснениями.

# Описание работы программы:

1. Подключение необходимых заголовочных файлов и пространств имён.

2. Определение начальных значений для интервала и ограничения количества итераций.

3. Определение класса FindRoot, который содержит параметры и методы для поиска корней и вычисления интегралов.

4. Определение структуры Method внутри класса FindRoot, которая содержит названия различных методов.

5. Реализация метода input\_EPS для ввода точности вычисления корней.

6. Перегрузка оператора << для вывода информации о поиске корней на экран.

7. Определение главной функции main.

8. Вызов функции input\_interval для ввода интервала.

9. Создание объекта FindRoot и объекта Integral.

10. Выбор функции для вычислений с помощью объекта FunctionSelector.

11. Вывод на экран шапки с цветным текстом.

12. Выполнение поиска корней различными методами и вывод результатов на экран.

13. Выполнение вычисления интегралов различными методами и вывод результатов на экран.

Программа предоставляет возможность пользователю искать корни уравнений и вычислять определенные интегралы. В начале программы пользователю предлагается ввести интервал, на котором будут производиться вычисления. Затем пользователю предлагается выбрать функцию, для которой будет выполняться поиск корней и вычисление интегралов. Программа поддерживает несколько методов для поиска корней, включая метод Bisect, LineSearch, Chord, Tangent и Combine. Для вычисления интегралов также реализованы различные методы, включая метод прямоугольников, метод трапеций, метод Симпсона и метод Монте-Карло.

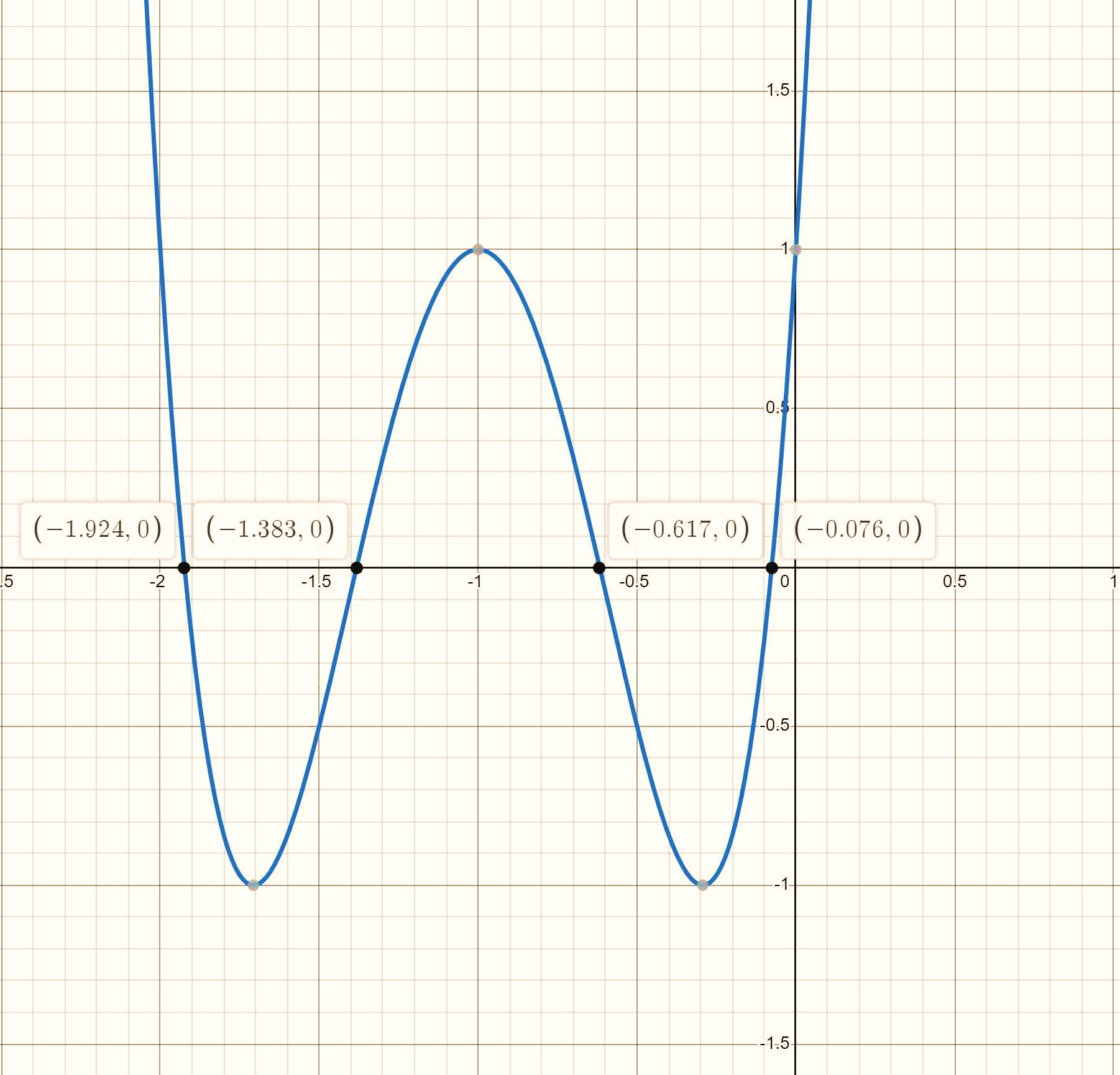
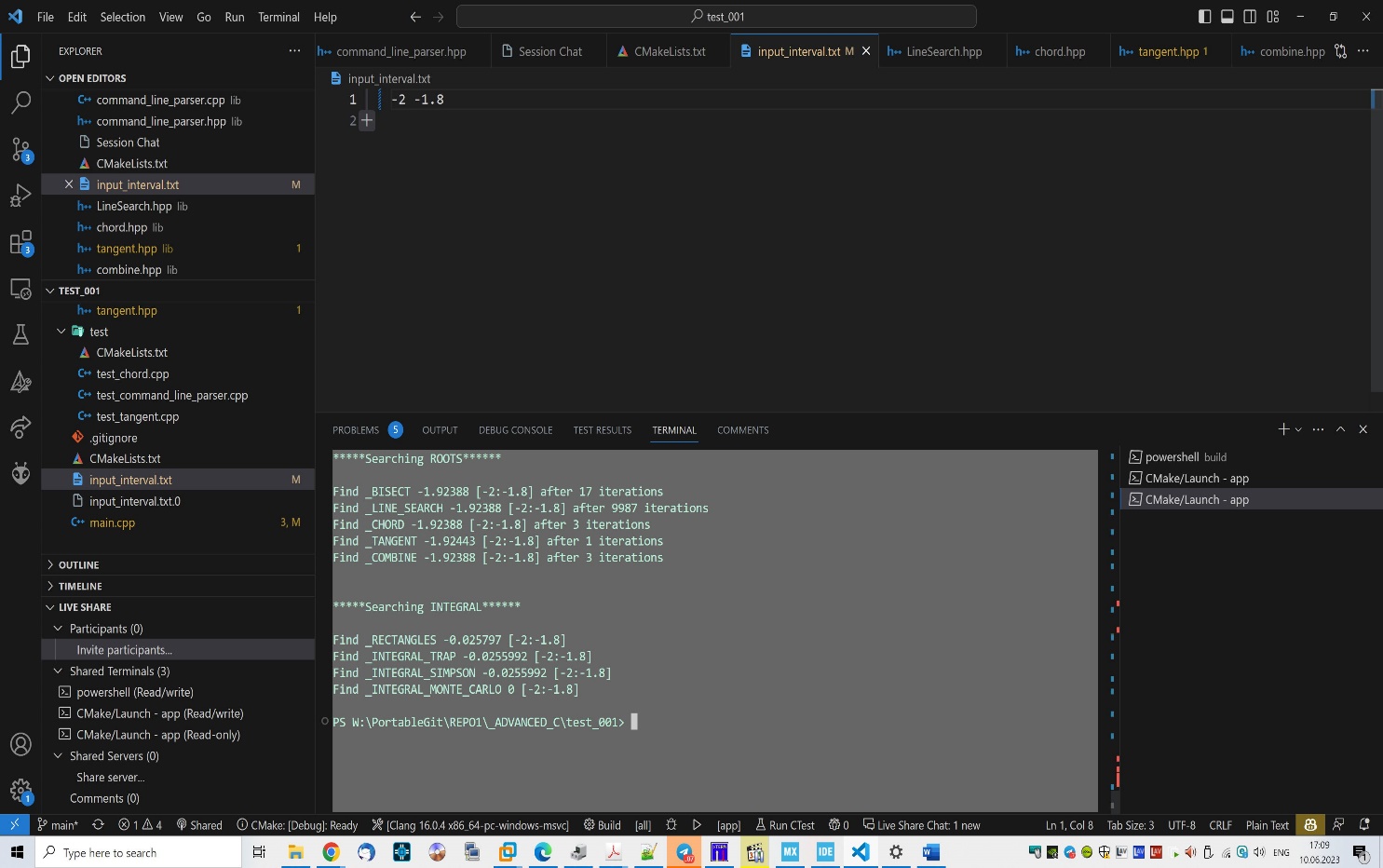
После выбора функции и методов, программа выполняет поиск корней с использованием выбранных методов и вычисляет интегралы с использованием выбранных методов. Результаты поиска корней и вычисления интегралов выводятся на экран. Программа имеет ограничение на количество итераций для каждого метода и точность вычислений, которую пользователь может ввести самостоятельно.

Рисунок 1 Корни уравнения 8𝑥^4 + 32𝑥^3 + 40𝑥^2 + 16𝑥 + 1

Программа также предоставляет возможность красочного вывода результатов на экран, используя коды ANSI для управления цветом текста. Это делает вывод более наглядным и удобочитаемым.

Для проверки правильности расчетов корней уравнения было произведено построение графика функции 8𝑥^4 + 32𝑥^3 + 40𝑥^2 + 16𝑥 + 1. Рис1. На основании построенного графика в программу было введен интервал поиска самого отрицательного корня со значением -1.924. Интервал был выбран от -2.0 до -1.8. На рис. 2 показан результат расчета этого корня различными методами. Как можно легко убедиться, данные численных методов расчета дают такой же результат, как и полученный графическим методом.

Рисунок 2 Корни уравнения 8𝑥^4 + 32𝑥^3 + 40𝑥^2 + 16𝑥 + 1



В рамках курсовой работы необходимо рассчитать площадь фигуры, образованной пересечением трех кривых, изображенных на рис3.

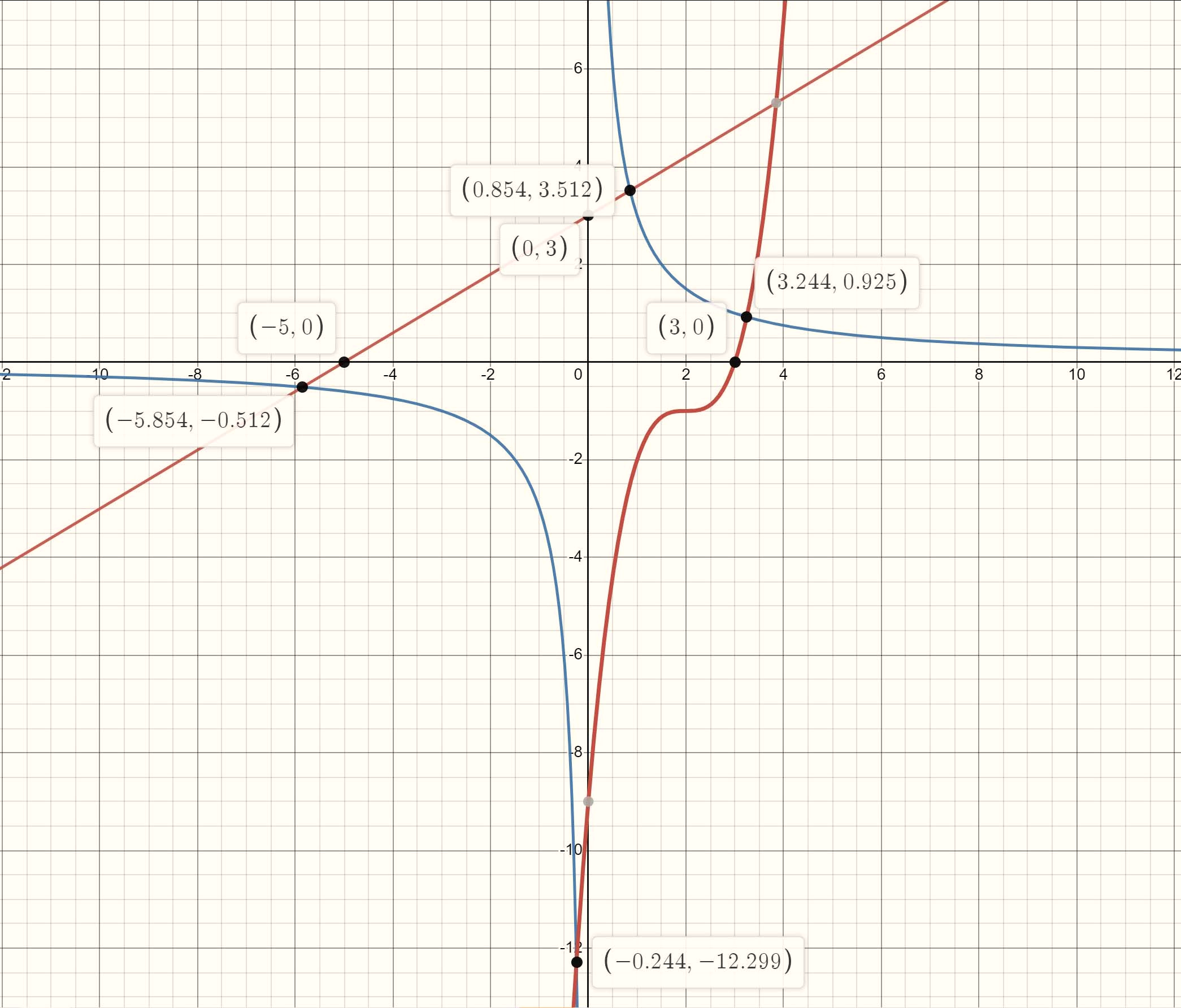
Искомая площадь ограничена прямой 𝑓1 = 0. 6𝑥 + 3, полиномом 3 порядка 𝑓2 = (𝑥 − 2)^3 – 1 и гиперболой 𝑓3 = 3 / 𝑥. Для расчета площади разобьем фигуру, ограниченную этими кривыми на отдельные части и расчитаем площадь этих частей численными методами.

Рисунок 3 Графики трех функций со значением корней и координатами точек пересечения

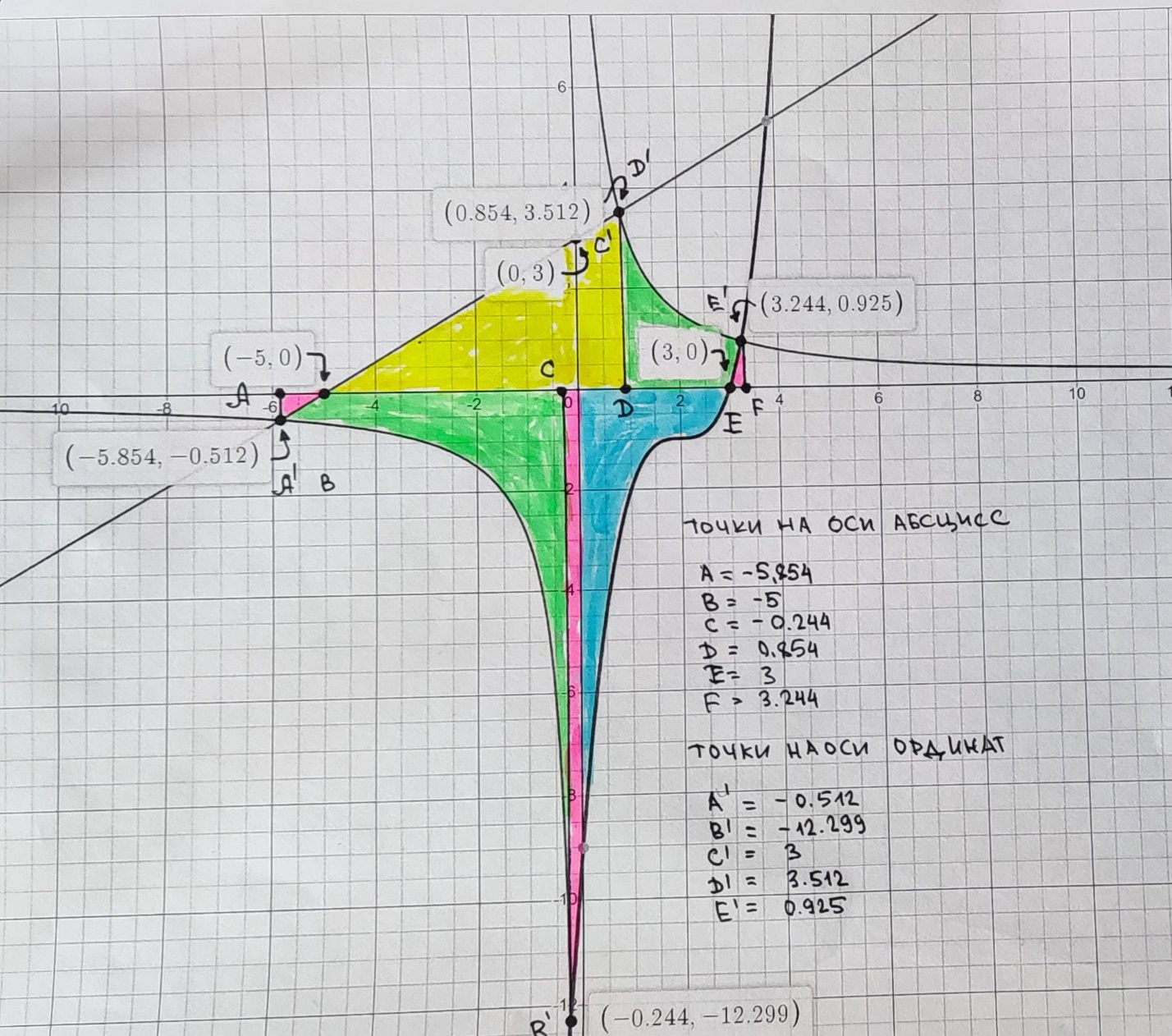


Рисунок 4 Графики трех функций со значением корней и координатами точек пересечения. Цветом показаны площади элементарных фигур, составляющих искомую площадь.

Анализ разбиения искомой площади показал, что по крайней мере три фигуры являются прямоуголиными треугольниками и их площадь может быть определена простым перемножением длин катетов деленные пополам.

Так площадь треугольника А’АВ равна (5.854-5)\*0.512/2=0.2186, треугольника образованного вершинами ВDD’ 5.854\*3.512/2=10.28, а треугольника ЕFE’ 0.244\*0.925/2=0.1129. При этом в окончательную площадь треугольники А’АВ и ЕFE’ входят с отрицательными знаками.

Остальные площади будут вычислены численными методами.

Площадь фигуры ограниченной отрицательной ветвью гиперболы, ось абсцисс с вершинами в точках А’АВ’C. Пределы интегрирования от АС.