Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Рязанский государственный радиотехнический университет имени В.Ф. Уткина»

Кафедра «Вычислительная и прикладная математика»

**Отчет по лабораторной работе № 2**

Решение уравнений с одной переменной

по дисциплине: «Вычислительная математика»

Выполнил:  
студент группы 143  
Попов К.И.  
Проверил:   
доц. каф. ВПМ  
Крошилина С.В.

# Рязань 2024

# 1 Задание (Вариант 2)

Найти положительный корень уравнения x5 + x – 0.2 = 0 с точностью ε = 0.0001 следующими методами: дихотомии; пропорциональных частей (хорд); касательных (Ньютона); модифицированным методом Ньютона; комбинированным методом; итерационным.

# 2 Блок-схемы алгоритмов

## 2.1 Метод дихотомии

Схема метода дихотомии представлена на рисунке 1.

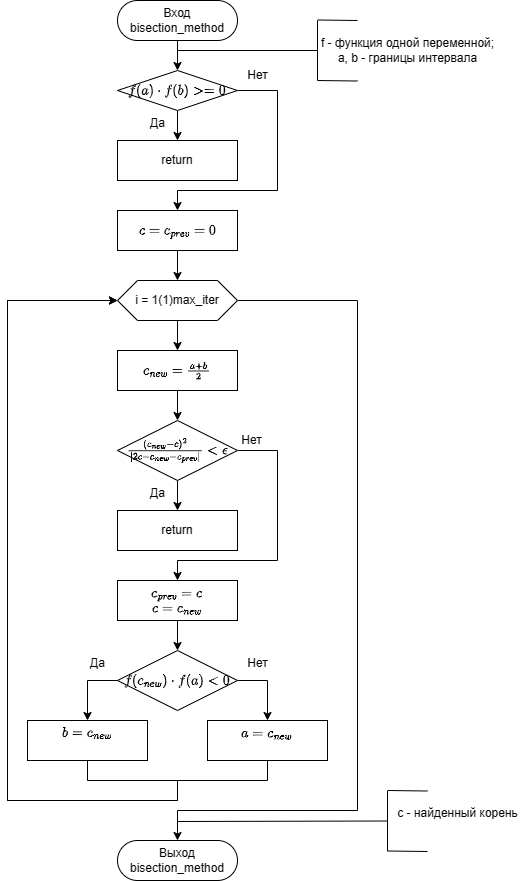


Рисунок 1 – Схема метода дихотомии

## 2.2 Метод хорд

Схема метода дихотомии представлена на рисунке 2.

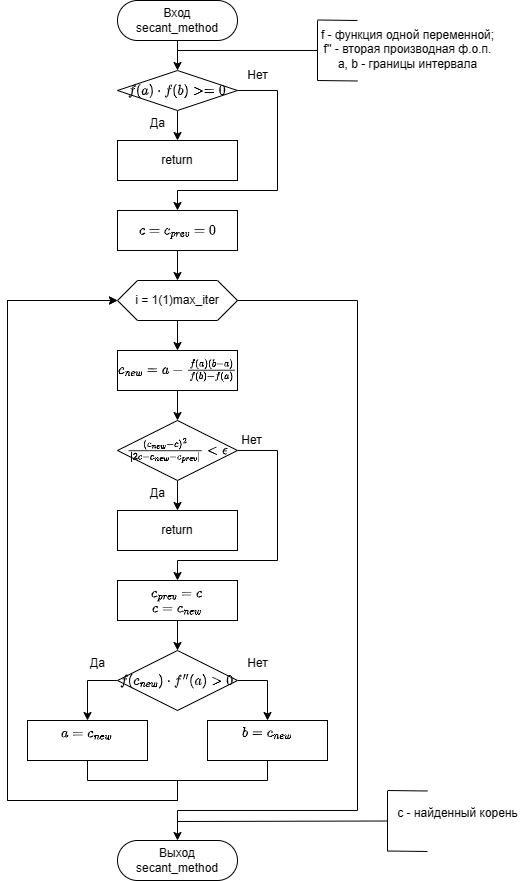


Рисунок 2 – Схема метода хорд

## 2.3 Метод Ньютона (касательных)

Схема метода Ньютона (касательных) представлена на рисунке 3.

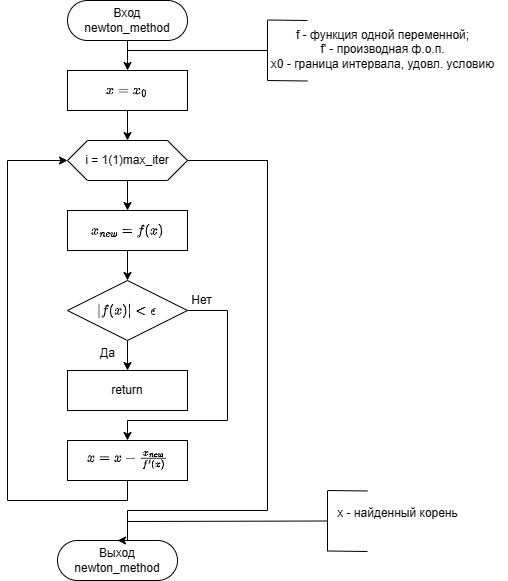


Рисунок 3 – Схема метода Ньютона (касательных)

## 2.4 Схема модифицированного метода Ньютона

Схема модифицированного метода Ньютона представлена на рисунке 4.

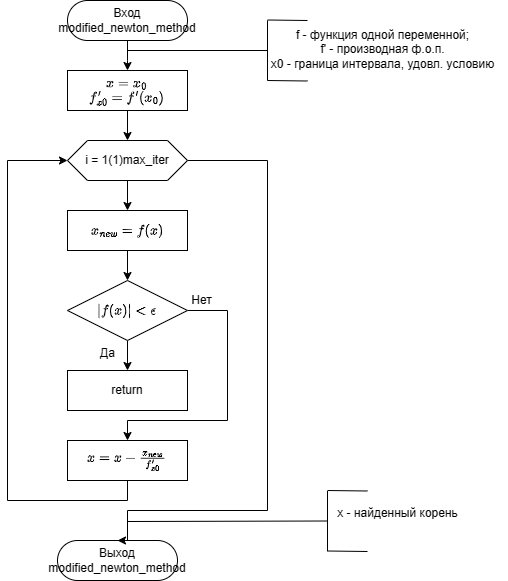


Рисунок 4 – Схема модифицированного метода Ньютона

## 2.5 Схема комбинированного метода

Схема модифицированного метода Ньютона представлена на рисунке 5.

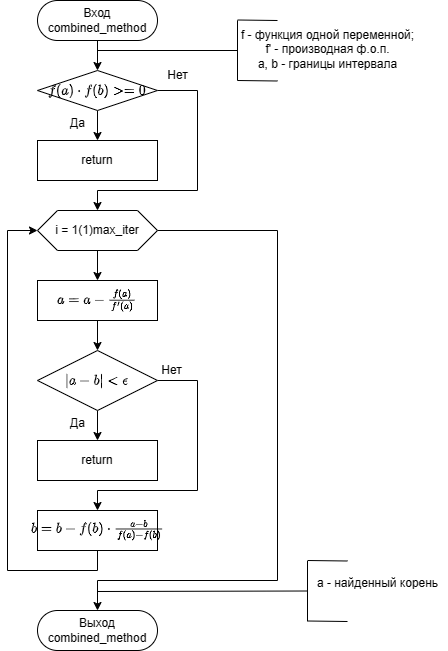


Рисунок 5 – Схема комбинированного метода

## 2.6 Схема итерационного метода

Схема итерационного метода Ньютона представлена на рисунке 6.

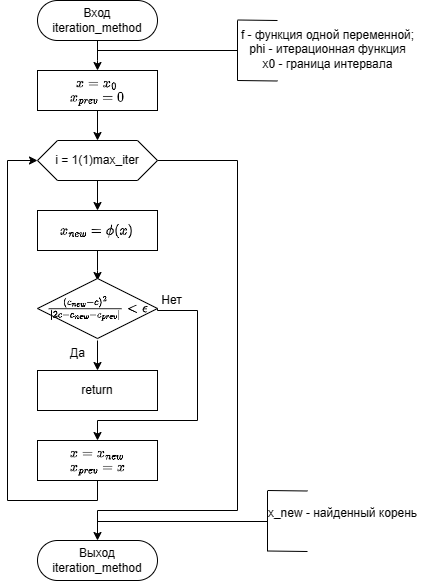


Рисунок 6 – Схема итерационного метода

# 3 Листинг алгоритмов

Реализация алгоритмов приведена на языке Python.

## 3.1 Листинг метода дихотомии

def bisection\_method(self, func, a, b, eps=1e-4, max\_iter=100):

        if func(a) \* func(b) >= 0:

            self.labelResult.setText("Невозможно найти корень на этом интервале.")

            return None

        c = c\_prev = 0

        for \_ in range(max\_iter):

            c\_new = (a + b) / 2

            if (c\_new - c)\*\*2 / abs(2 \* c - c\_new - c\_prev) < eps:

                self.labelResult.setText(f"Положительный корень: {c\_new}")

                return c\_new

            c\_prev, c = c, c\_new

            if func(c\_new) \* func(a) < 0:

                b = c\_new

            else:

                a = c\_new

        self.labelResult.setText("Достигнуто максимальное количество итераций.")

        return None

## 3.2 Листинг метода хорд

def secant\_method(self, func, second\_derivative, a, b, eps=1e-4, max\_iter=100):

        if func(a) \* func(b) >= 0:

            self.labelResult.setText("Невозможно найти корень на этом интервале.")

            return None

        c = c\_prev = 0

        for \_ in range(max\_iter):

            c\_new = a - (func(a) \* (b - a)) / (func(b) - func(a))

            if (c\_new - c)\*\*2 / abs(2 \* c - c\_new - c\_prev) < eps:

                self.labelResult.setText(f"Положительный корень: {c\_new}")

                return c\_new

            c\_prev, c = c, c\_new

            if func(c\_new) \* second\_derivative(a) > 0:

                a = c\_new

            else:

                b = c\_new

        self.labelResult.setText("Достигнуто максимальное количество итераций.")

        return None

## 3.3 Листинг метода Ньютона (касательных)

def newton\_method(self, func, derivative, x0, eps=1e-4, max\_iter=100):

        x = x0

        for \_ in range(max\_iter):

            x\_new = func(x)

            if abs(x\_new) < eps:

                self.labelResult.setText(f"Положительный корень: {x}")

                return x

            x = x - x\_new / derivative(x)

        self.labelResult.setText("Достигнуто максимальное количество итераций.")

        return None

## 3.4 Листинг модифицированного метода Ньютона

def modified\_newton\_method(self, func, derivative, x0, eps=1e-4, max\_iter=100):

        x = x0

        derivative\_x0 = derivative(x0)

        for \_ in range(max\_iter):

            x\_new = func(x)

            if abs(x\_new) < eps:

                self.labelResult.setText(f"Положительный корень: {x}")

                return x

            x = x - x\_new / derivative\_x0

        self.labelResult.setText("Достигнуто максимальное количество итераций.")

        return None

## 3.5 Листинг комбинированного метода

def combined\_method(self, func, derivative, a, b, eps=1e-4, max\_iter=100):

        if func(a) \* func(b) >= 0:

            self.labelResult.setText("Невозможно найти корень на этом интервале.")

            return None

        for \_ in range(max\_iter):

            a = a - func(a) / derivative(a)

            if abs(a - b) < eps:

                self.labelResult.setText(f"Положительный корень: {a}")

                return a

            b = b - func(b) \* (a - b) / (func(a) - func(b))

        self.labelResult.setText("Достигнуто максимальное количество итераций.")

        return None

## 3.6 Листинг итерационного метода

def iteration\_method(self, phi, x0, eps=1e-4, max\_iter=100):

        x, x\_prev = x0, 0

        for \_ in range(max\_iter):

            x\_new = phi(x)

            if (x\_new - x)\*\*2 / abs(2 \* x - x\_new - x\_prev) < eps:

                self.labelResult.setText(f"Положительный корень: {x\_new}")

                return x\_new

            x, x\_prev = x\_new, x

        self.labelResult.setText("Достигнуто максимальное количество итераций.")

        return None

# 4 Пользовательский интерфейс

На рисунке 7 показано главное пользовательское окно взаимодействия с программой, в котором пользователь может получить корень уравнения выбранным методом.

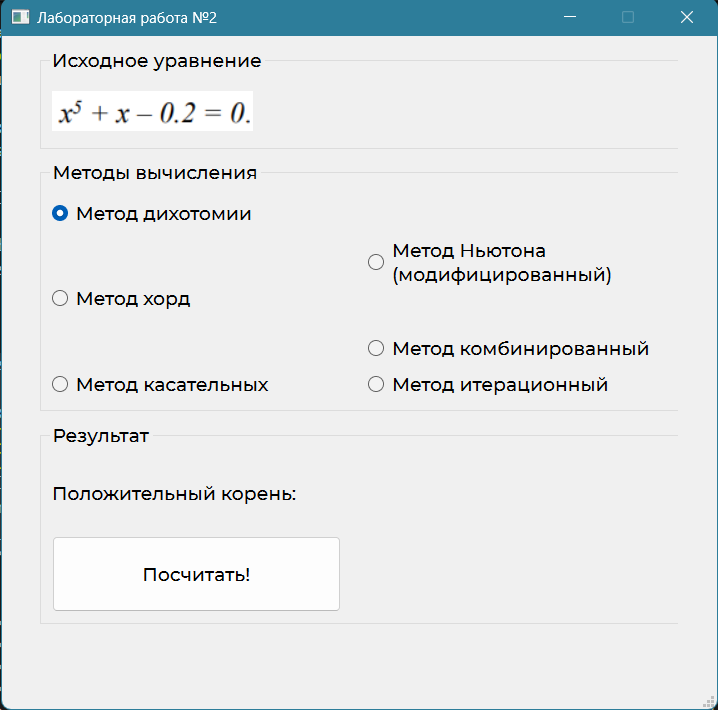


Рисунок 7 – Главное окно

# 5 Результат работы программы

## 5.1 Результат метода дихотомии

Ответ, посчитанный методом дихотомии, представлен на рисунке 8.

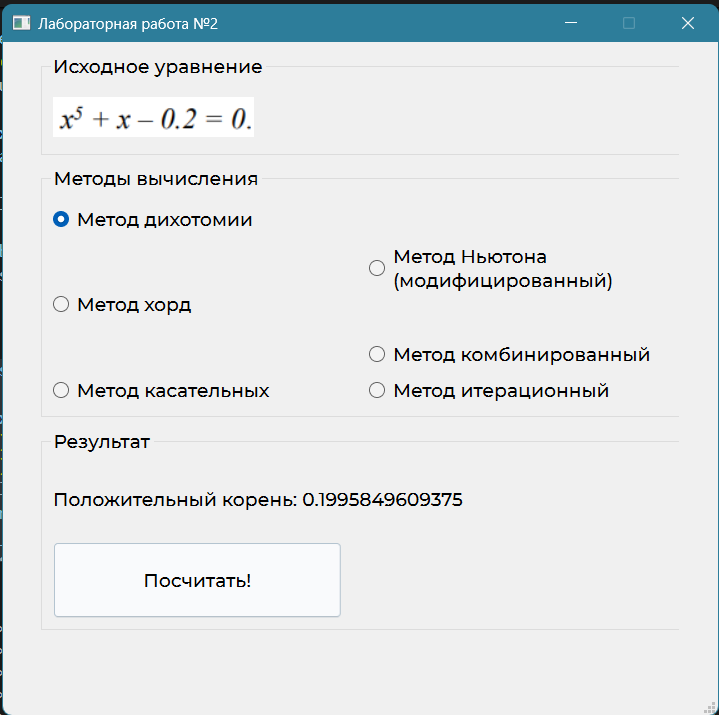


Рисунок 8 – Результат работы метода дихотомии

## 5.2 Результат метода хорд

Ответ, посчитанный методом дихотомии, представлен на рисунке 9. Вторая производная, задействованная в методе: .

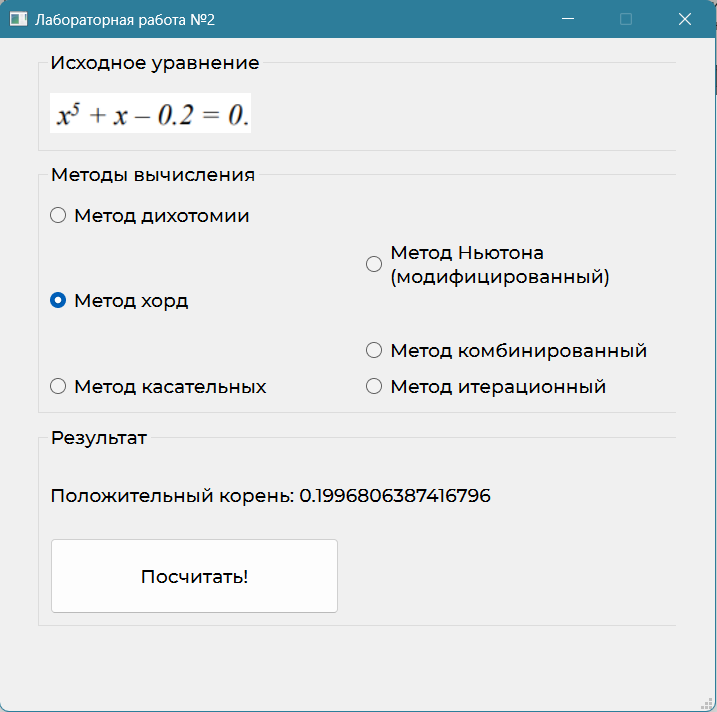


Рисунок 9 – Результат работы метода хорд

## 5.3 Результат метода хорд

Ответ, посчитанный методом Ньютона (касательных), представлен на рисунке 10. Производная функции: .

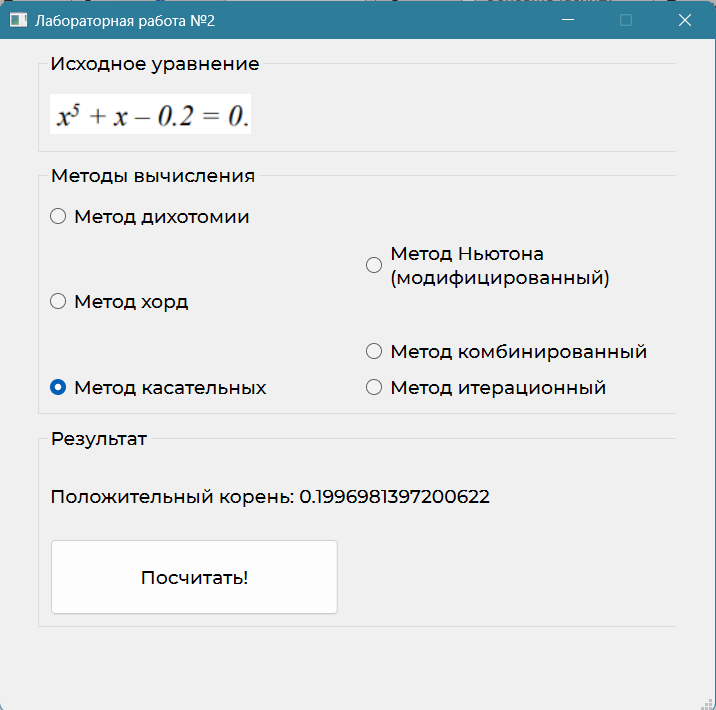


Рисунок 10 – Результат работы метода Ньютона (касательных)

## 5.4 Результат модифицированного метода Ньютона

Ответ, посчитанный модифицированным методом Ньютона, представлен на рисунке 11.

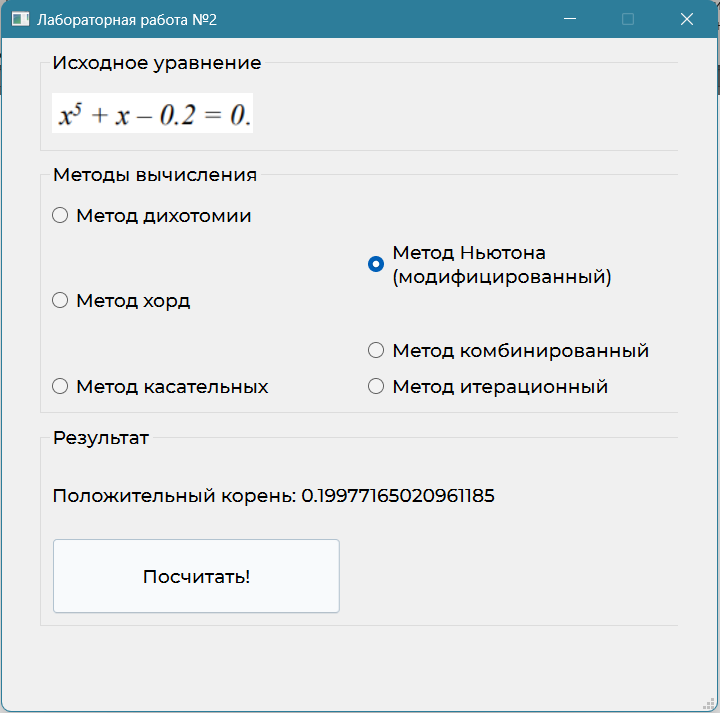


Рисунок 11 – Результат работы модифицированного метода Ньютона

## 5.5 Результат комбинированного метода

Ответ, посчитанный комбинированным методом, представлен на рисунке 12.

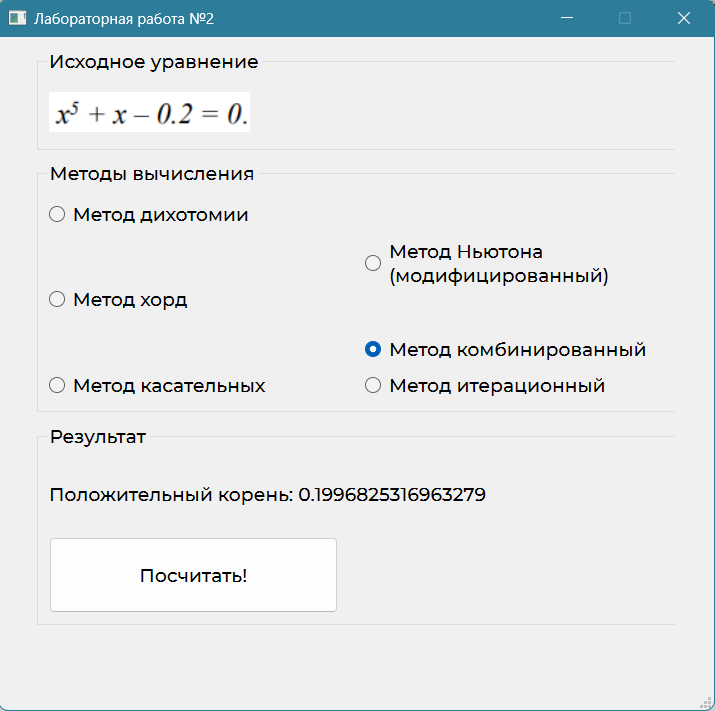


Рисунок 12 – Результат работы комбинированного метода

## 5.6 Результат итерационного метода

Ответ, посчитанный комбинированным методом, представлен на рисунке 13. В качестве вспомогательной функции была выбрана

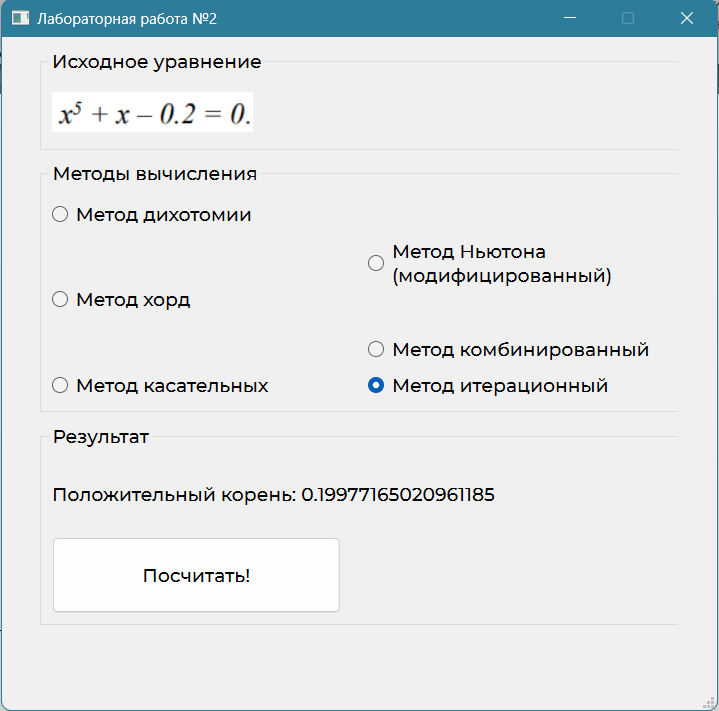


Рисунок 13 – Результат работы итерационного метода

# 6 Проверка полученного результат

График функции одной переменной представлен при помощи интернет-сервиса Desmos на рисунке 14.

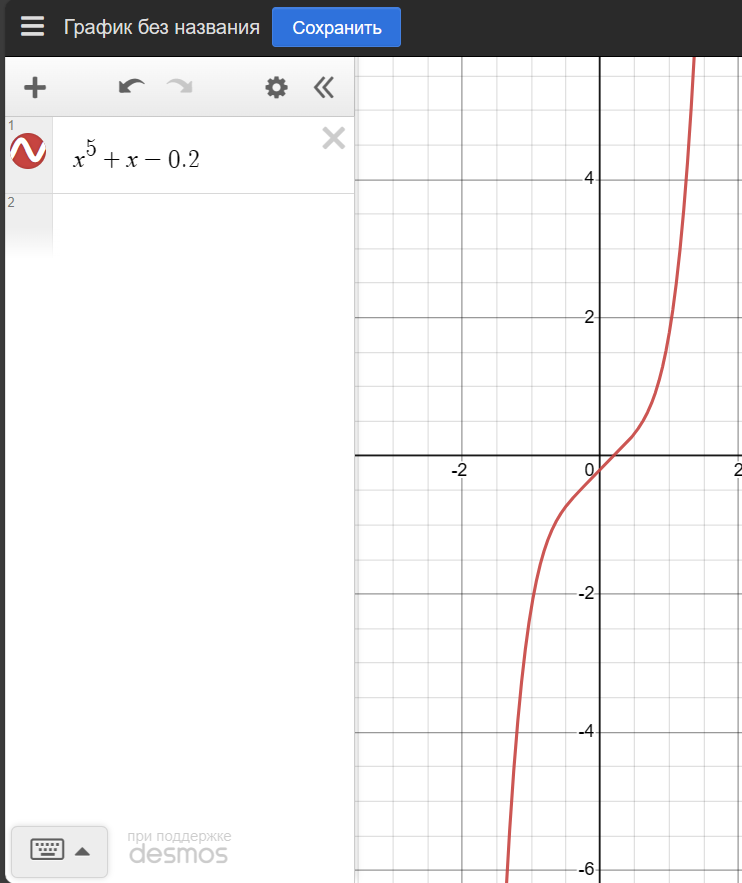


Рисунок 14 – График заданной функции

Из графического анализа видно, что корнем для уравнения f(x) = 0 является единственное действительное значение. Интервалом уточнения корня используется отрезок [0; 1].

Проверка уточненного корня произведена при помощи интернет-сервиса WolframAlpha (Рисунок 15).

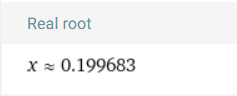
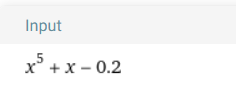


Рисунок 15 – Проверка корня уравнения