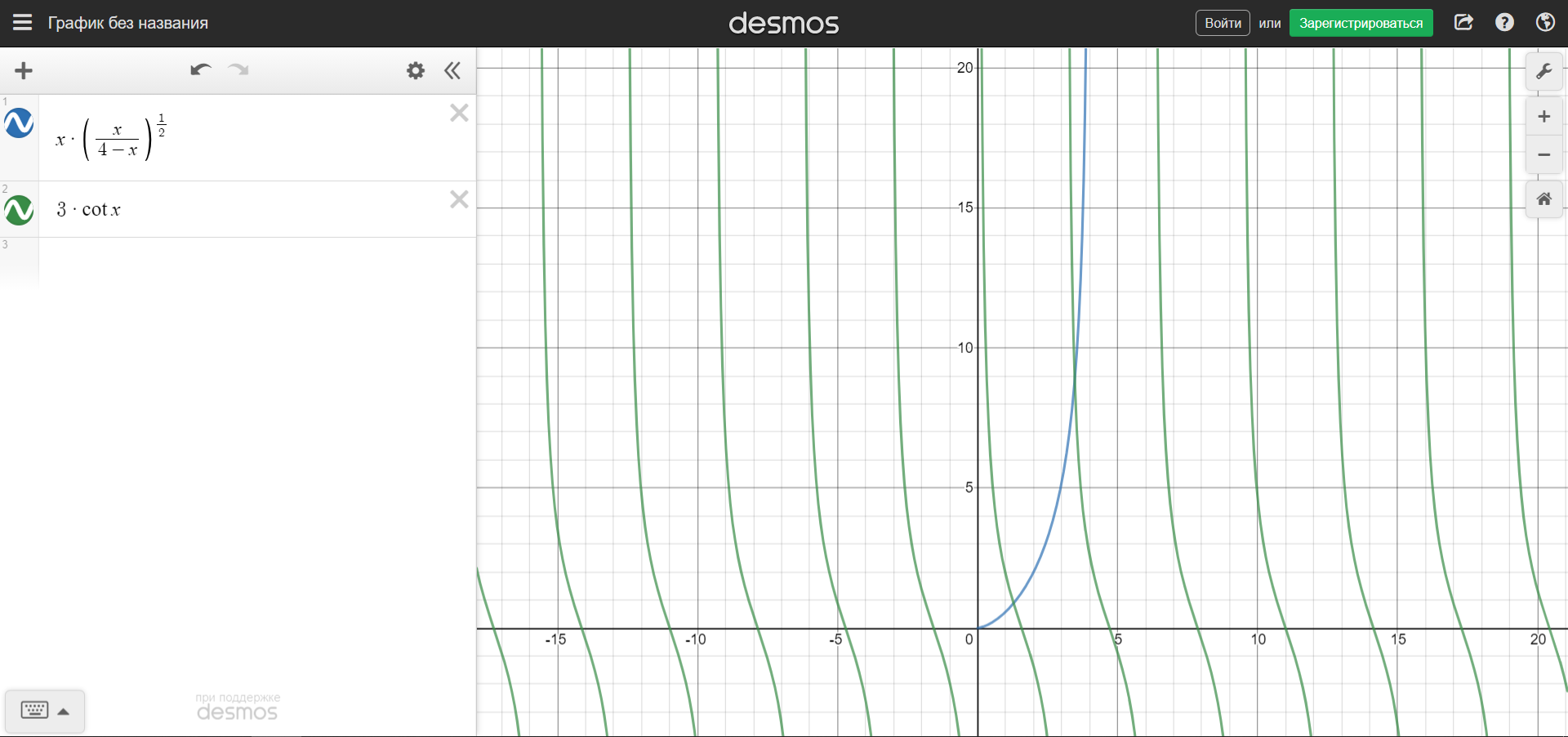
**Цель:** рассмотреть 4 метода решения уравнения

1. Бисекция;
2. Метод секущих;
3. Метод ложной позиции;
4. Метод Ньютона

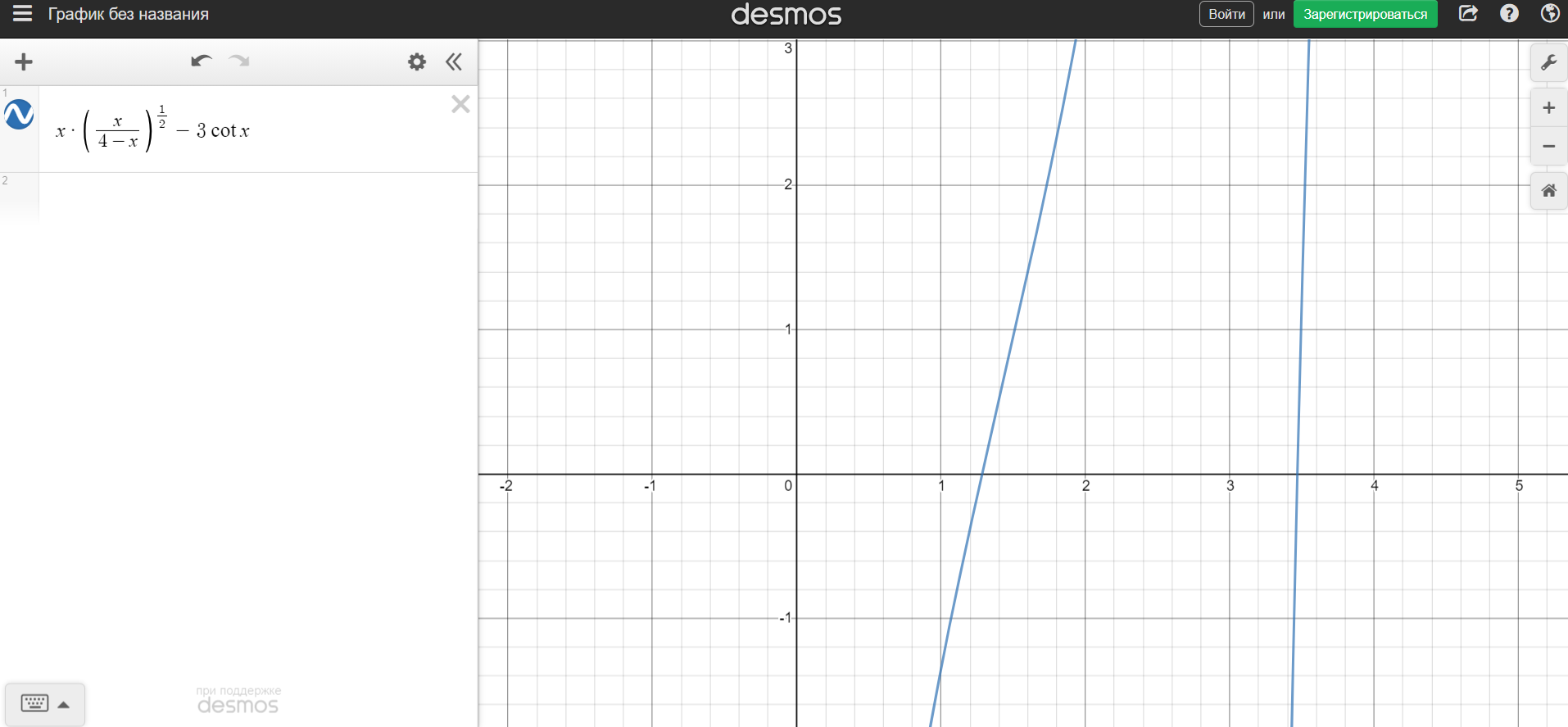
Для данных методов следует задать следующие точности: 0.01, 0.001, 0.0001. Для каждого метода и каждой точности найти и составить список итераций. Написать вывод о скорости сходимости данных методов решений уравнения.

**Рассматриваемое уравнение:**

Построим графики этих функций с помощью программы Desmos:



Построим график функции:



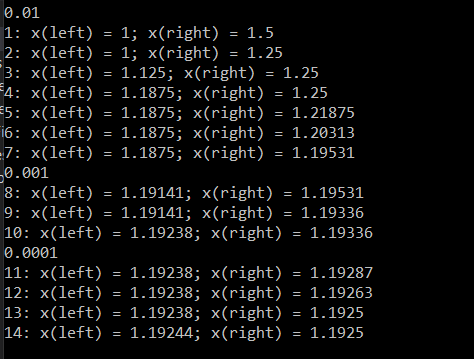
Данное уравнение имеет всего 2 корня на отрезке от [1,4]. В лабораторной работе будем рассматривать только один корень на отрезке от [1,2].

В качестве среды разработки была выбрана Visual Studio 2015. Язык разработки – C++.

**Метод бисекции**

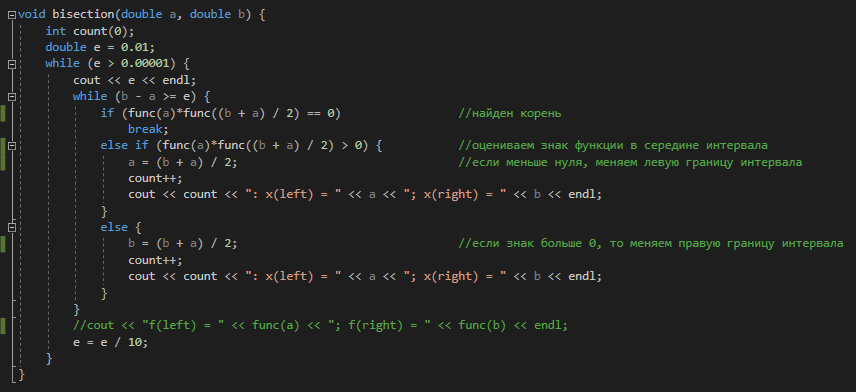
Идея метода бисекции проста. Известно, что на некотором интервале функция проходит через ноль, поскольку меняет знак. Необходимо оценить функцию в середине интервала и проверить её знак. Используем среднюю точку, чтобы изменить границу интервала с тем же знаком. После каждой интеракции интервал уменьшается в два раза. Остановкой алгоритма считается достигнутая точность. В нашем случае, когда алгоритм достигнет точности 0.0001, он остановит работу и корнем уравнения будет считаться точка равная , где – левая граница интервала, – правая граница интервала.

**Результат работы алгоритма:**



Точность 0.01 была достигнута на 7-ой итерации, 0.001 на 10-ой и точность 0.0001 на 14-ой итерации.

**Код алгоритма:**

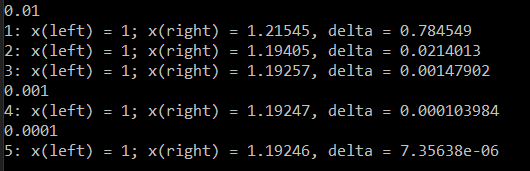


**Метод ложной позиции**

В методе ложной позиции нелинейная функция g(x) заменяется линейной функцией f(x) в интервале (x0, x1), и берется корень линейной функции f(x), x = c как следующее приближение корня нелинейной функции g(x), x = a. Корень линейной функции f(x), то есть x = c, не является корнем данной нелинейной функции g (x). Это ложная позиция, которая и дает название методу. Теперь у нас есть два интервала (x0, c) и (c, x1). Как и в бисекции, выбираем интервал, на котором у функции меняется знак. Следующее приближение вычисляется по формуле:

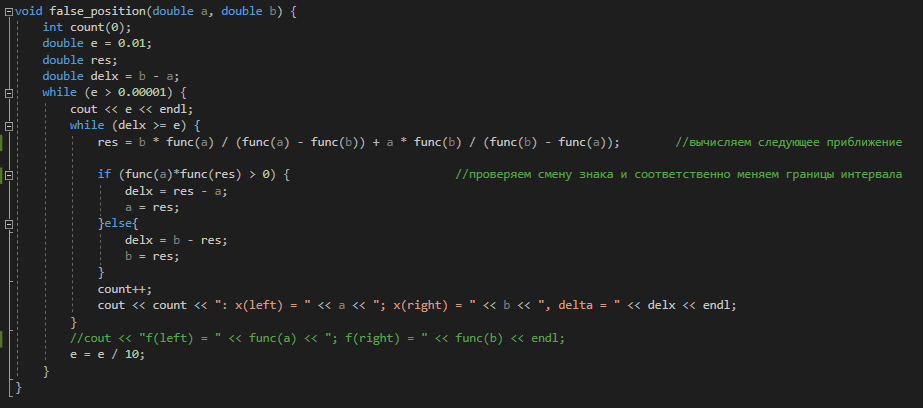
Начальные точки x0, x1 такие, что g(x0)\*g(x1) < 0, находятся раньше, т.е. заданы (как методом бисекции). Алгоритм выполняется пока разница между двумя последовательными приближениями не станет меньше заданной точности.

**Результат работы алгоритма:**



Точность 0.01 была достигнута на 3-ей итерации, точность 0.001 на 4-ой, а точность 0.0001 на 5-ой.

**Код алгоритма:**

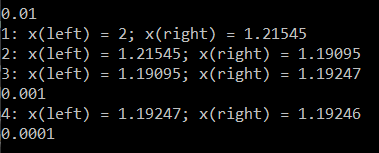
****

**Метод секущих**

Для функций, гладких около корня, метод ложной позиции и метод секущей, обычно сходятся быстрее, чем бисекция. В обоих этих методах предполагается, что функция приблизительно линейна в локальной области интереса, и следующее улучшение корня берется в точке, где приближенная линия пересекает ось. После каждой итерации одна из предыдущих граничных точек отбрасывается в пользу последней оценки корня. Следующее приближение вычисляется по формуле, используемой в методе ложной позиции:

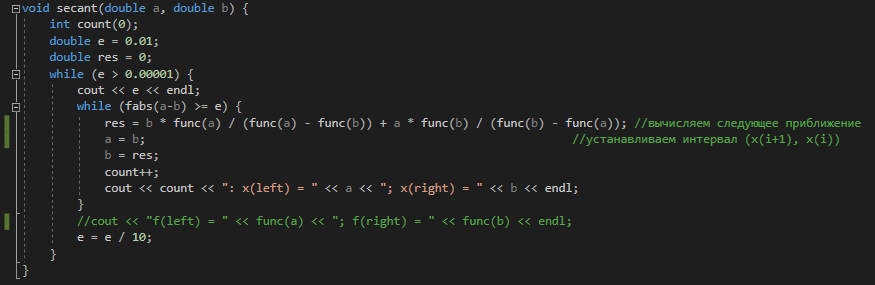
Единственное, но очень существенное, различие между методом секущей и методом ложной позиции можно описать следующим образом: среди двух интервалов (xi-1, xi+1) и (xi+1, xi) метод секущих всегда сохраняет интервал (xi+1, xi), образованный двумя последними точками, независимо от знаков рассматриваемой функции g(x) на концах. При этом метод ложной позиции сохраняет как раз интервал, на концах которого g(x) имеет разные знаки.

**Результат работы алгоритма:**

****

Точность 0.01 была достигнута на 3-ей итерации, а точность 0.001, как и точность 0.0001, на 4-ой итерации.

**Код алгоритма:**

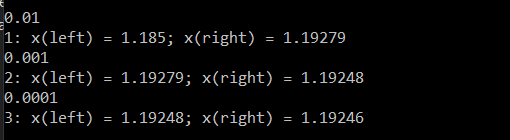


**Метод Ньютона**

В отличие от предыдущих методов, метод Ньютона используют только одну точку в качестве начальных данных. Вторая точка интервала вычисляется с помощью следующей формулы:

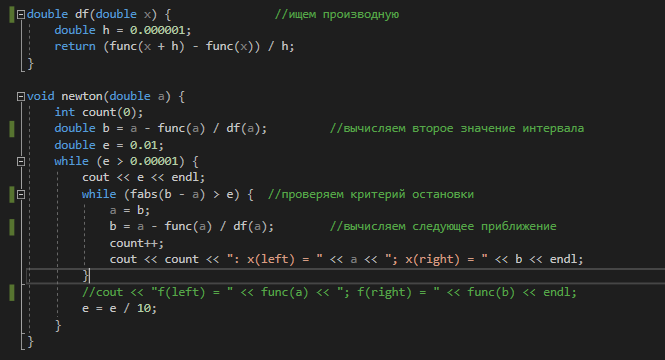
Где g’(xi) – производная функции. Критерий остановки алгоритма, как и в предыдущих методах, определяется близостью двух последовательных приближений корня.

**Результат работы программы:**



Точность 0.01 достигнута на 1-ой итерации, точность 0.001 на 2-ой, а точность 0.0001 на 3-ей итерации.

**Код алгоритма:**



**Вывод:**

Сравнительная таблица:

|  |  |  |
| --- | --- | --- |
| Название алгоритма | Количество итераций | Найденный корень (округляем до четырех знаков) |
| Бисекция | 14 | 1,1924 |
| Метод ложной позиции | 5 | 1,1924 |
| Метод секущих | 4 | 1,1924 |
| Метод Ньютона | 3 | 1,1924 |

Все методы справились с нахождением корня на заданном отрезке. Наиболее быструю скорость сходимости продемонстрировал метод Ньютона. Недостатками данного метода являются локальность и ограниченность на характер функции, а также вычисление производной на каждой итерации. Если функция сложна (аналитическое дифференцирование сопряжено со значительными трудностями), то используют метод секущих. Поскольку здесь не требуется нахождение производной. И при меньшей скорости сходимости, но при том же объеме вычислений, можно получить большую точность. Метод ложной позиции отличается от метода секущих только тем, что берутся не две последние точки, а те, что находятся вокруг корня. Метод бисекции применим для любой непрерывной функции. Он прост и надёжен, что безусловно является его достоинствами, однако он неприменим к ряду случаев (корни четной кратности, комплексные корни).

При выборе метода решения уравнения следует отталкиваться от начальных данных (одна точка интервала или две, характер функции) и от приоритета задачи (скорость сходимости, вычислительные затраты). Часто для решения уравнения применяют комбинации этих методов, сочетающие их достоинства.