Пермский национальный исследовательский политехнический университет

Кафедра “Информационные технологии и автоматизированные системы”

Лабораторная работа № 1

**Решение нелинейных уравнений**

Выполнил:

Студент РИС-24-2б Пугина Мария Сергеевна

Проверил:

Доцент кафедры ИТАС Ольга Андреевна Полякова

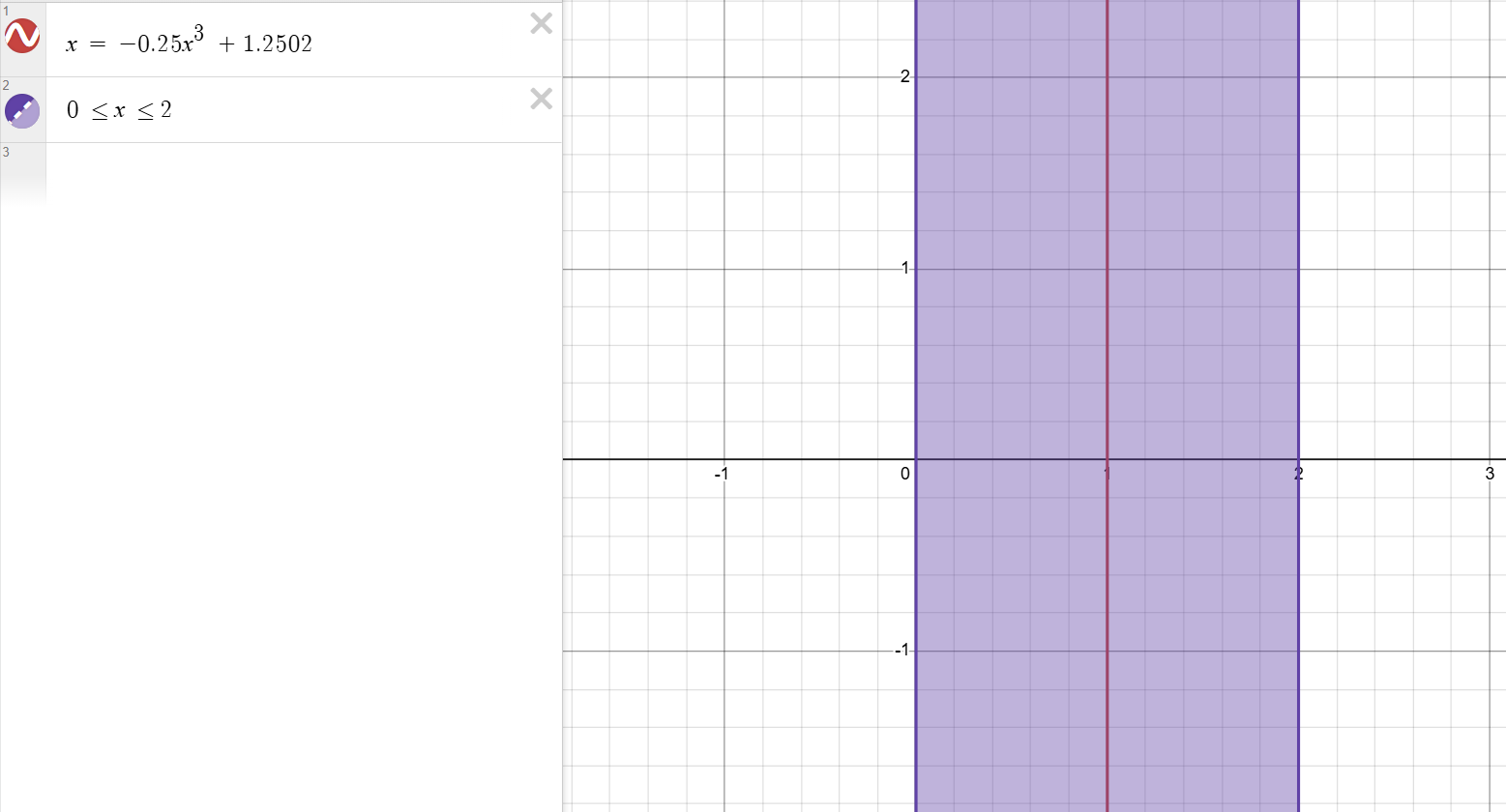
г. Пермь, 2024

Вариант 6

Постановка задачи: Разработать алгоритмы и написать программы на С++ для решения нелинейного уравнения.

Дано нелинейное уравнение: 0.25x3 + x – 1. 2502 = 0, отрезок, содержащий корень: [0;2], который содержит корень 1.0001. Epsilon (точность вычислений) = 10-6

График: x = -0.25x3 + 1.2502

Рисунок 1. Графическое решение уравнения.

Перед методами:

1. а - левый конец отрезка
2. b - правый конец отрезка
3. Epsilon - точность (10-6)
4. Обозначим функцию f(x) = 0.25x3 + x – 1. 2502.
5. Найдём первую производную от функции: f’(x) =
6. Найдём вторую производную от функции: f”(x) =
7. Проверим функцию на монотонность и непрерывность по формуле f(a) \* f”(a) > 0 или f(b) \* f”(b) > 0. По второй формуле получим, что

2.7498 \* 3 > 0.

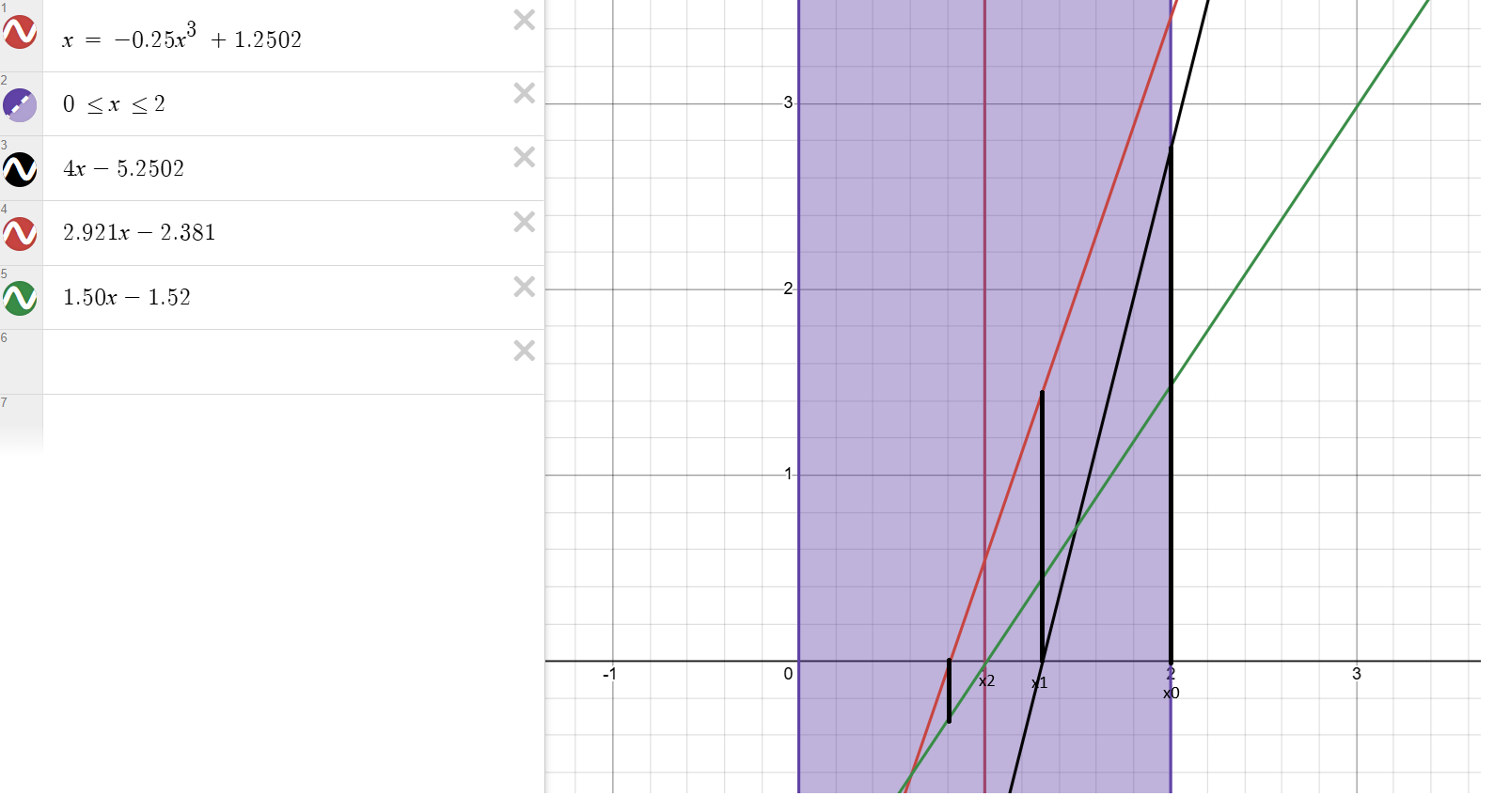
Метод Ньютона

1. Выберем из интервала [a; b] точку x0, которую будем считать исходным значением корня. Допустим x0 = b.
2. По методу касательных находится следующее значение корня - x1. Находим значение функции f(x0), то есть f(b).
3. Затем находим x1 по формуле xi-1 = xi - (f(xi))/(f’(xi))
4. После этого запускается цикл до тех пор, пока |x0 - x1| не станет меньше или равен заданной точности (10-6).

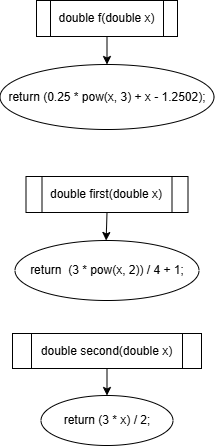
А) Значение x0 меняется на x1

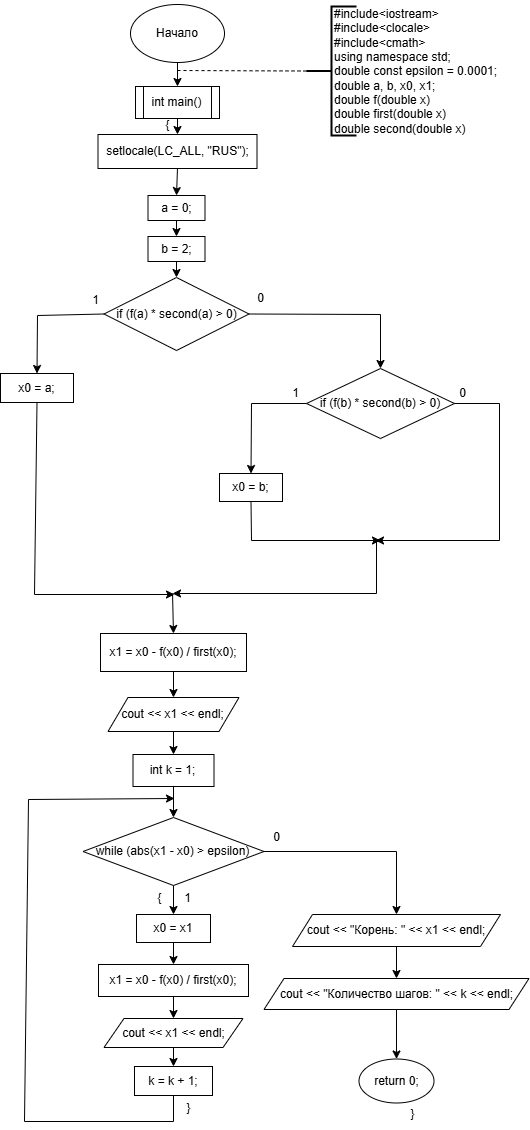
Б) Повторяется третий шаг

1. Выводится ответ.

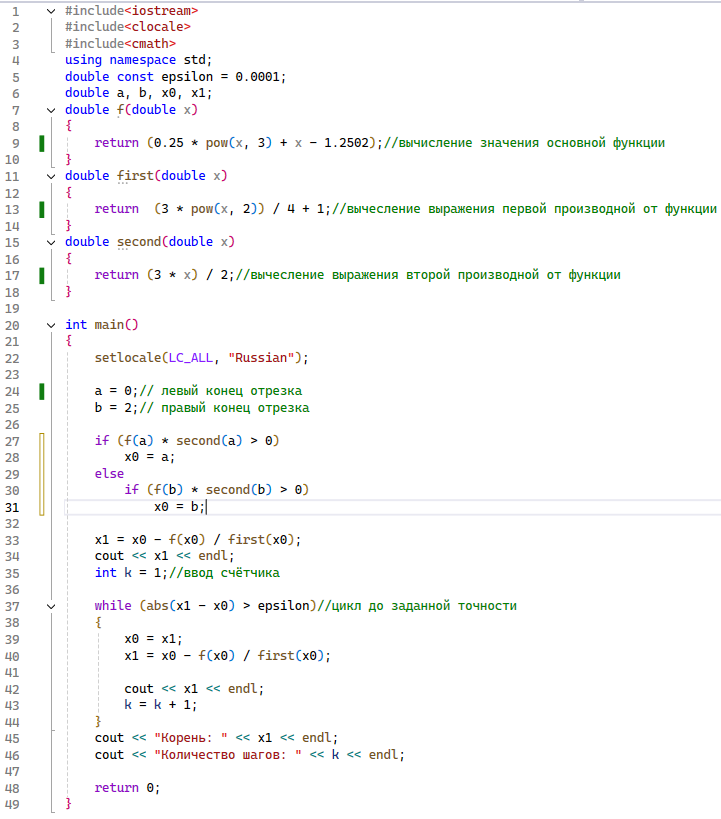


Блок-схема:

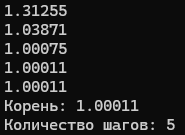




Код в С++:

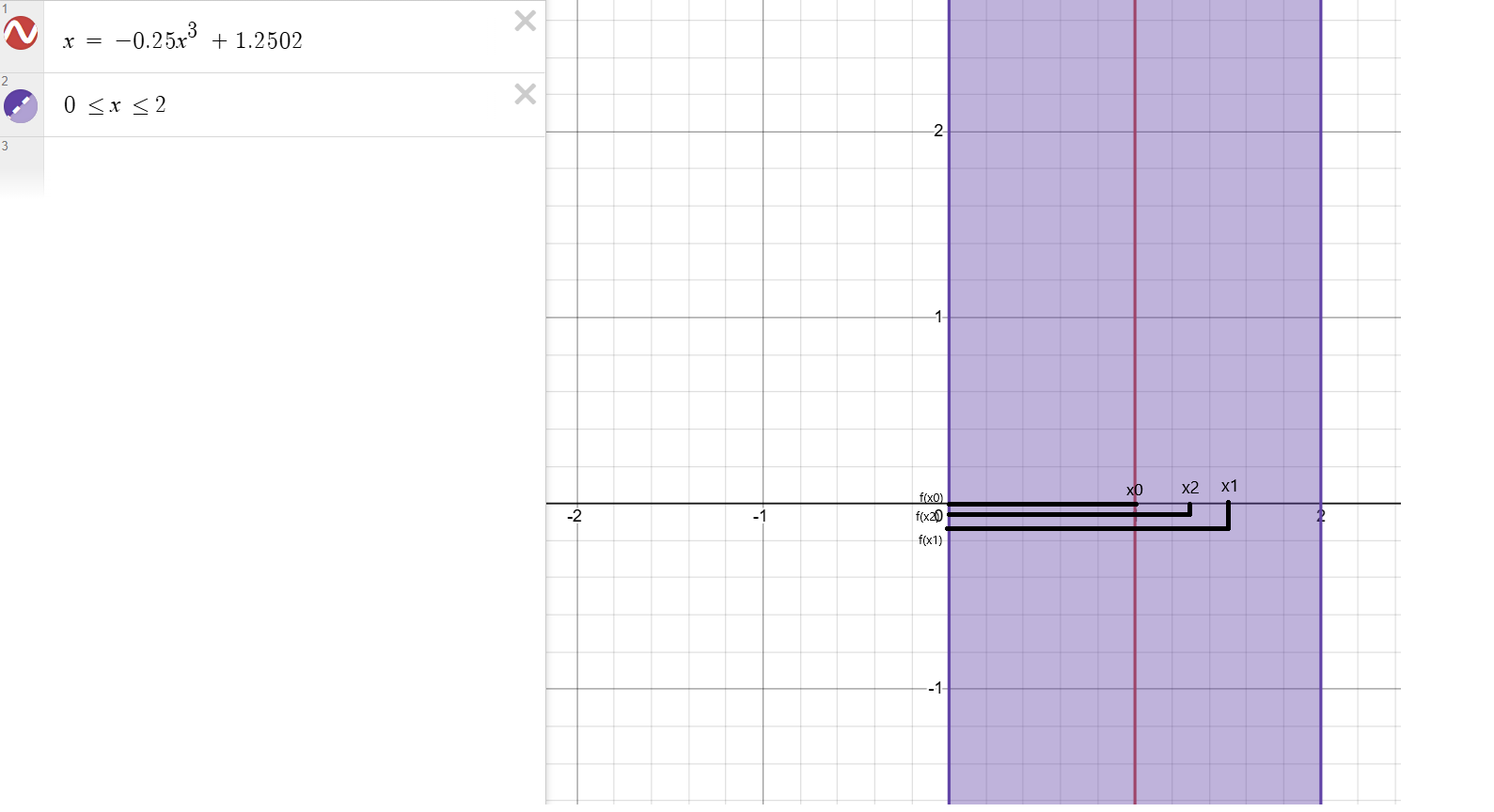


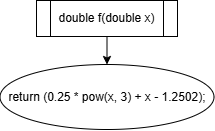
Отработка программы:

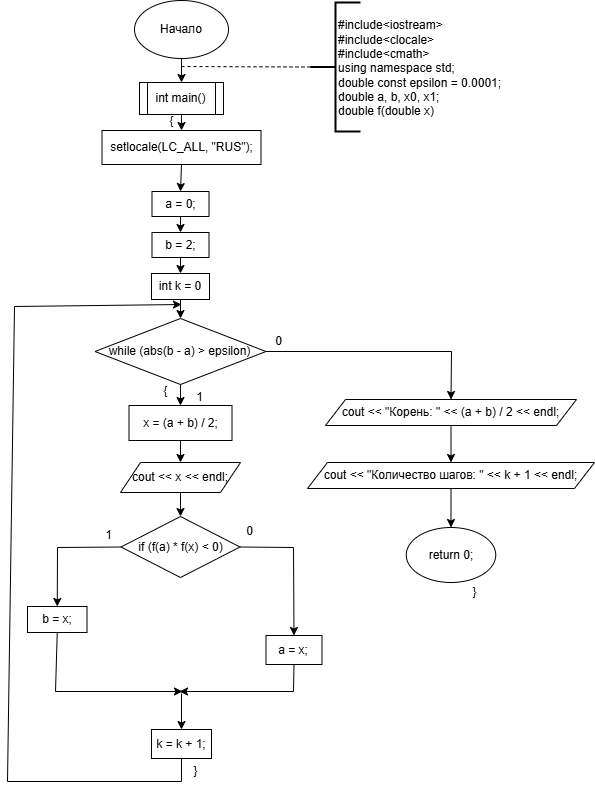


Метод половинного деления

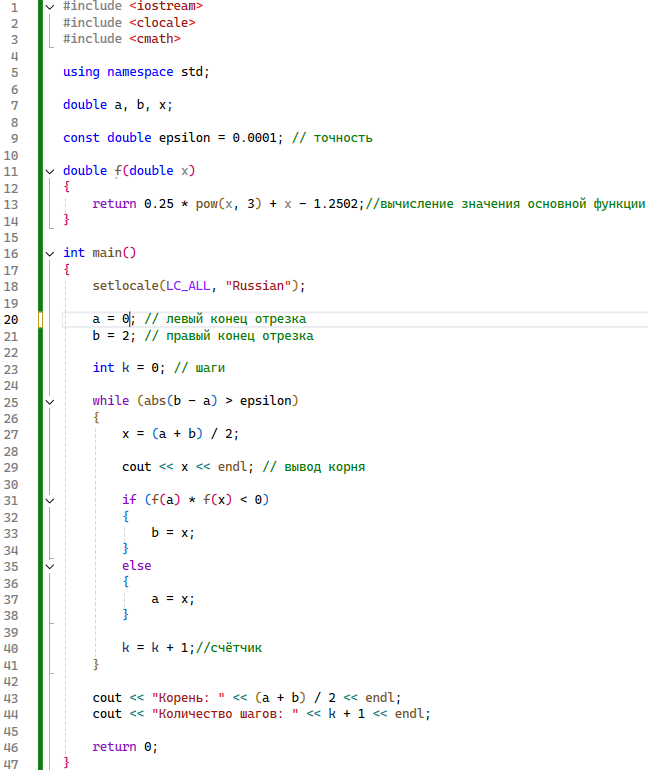
1. Предварительно надо проверить, что произведение f(a) \* f(b) < 0, то есть функция пересекает ось 0x в интервале [1: 2]. Получаем - - 3.4378 < 0.
2. Интервал [а; b] делится пополам x = (a + b) / 2. Полученная точка на половине интервала [a; b] считается начальным приближением корня.
3. Отбрасываем ту половину интервала, в которой не содержатся корни.
4. После этого запускается цикл, который продолжает работать до тех пор, пока разность между концами отрезка не станет меньше или равен Epsilon. |x1 – x0| < 10-6
5. Последний найденный xi и будет считаться корнем уравнения.

Блок-схема:

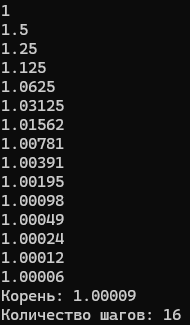




Код в С++:



Отработка программы:



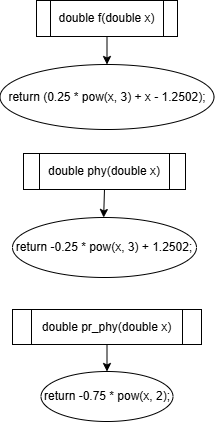
Метод итераций

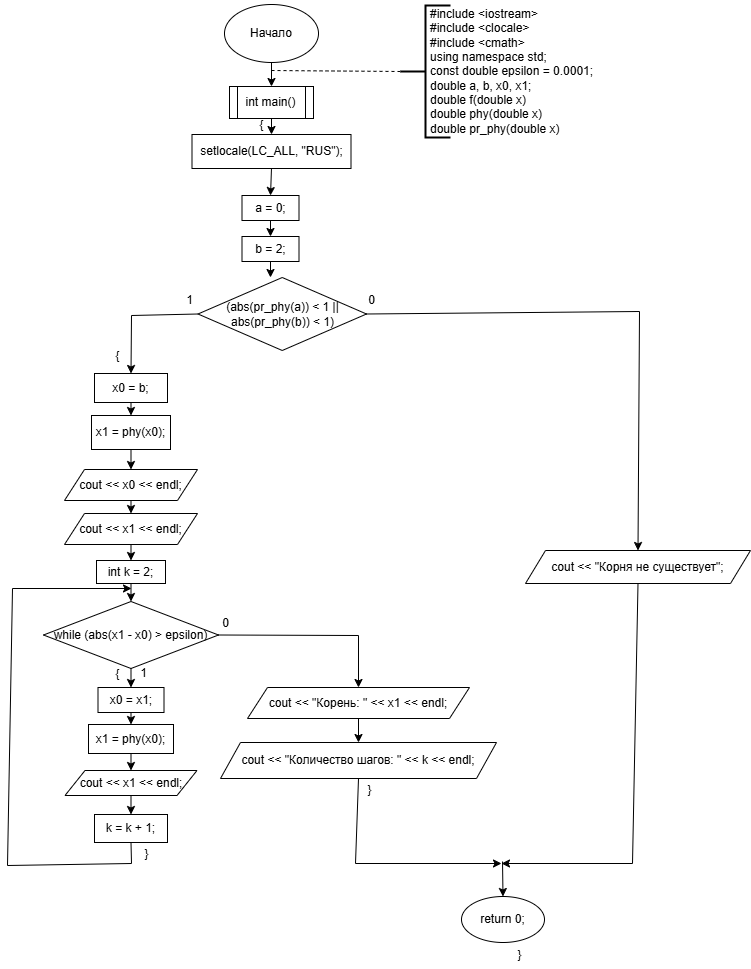
1. Выражаем вспомогательную функцию для дальнейших вычислений. Уравнение f(x) = 0 переписывается в виде x = φ(x). φ(x)=
2. Проверка сходимости. |φ’(x)| < 1. φ’(x) = .

φ’(2) = 0.6249. |0.6249| < 1. Это значение удовлетворяет сходимости.

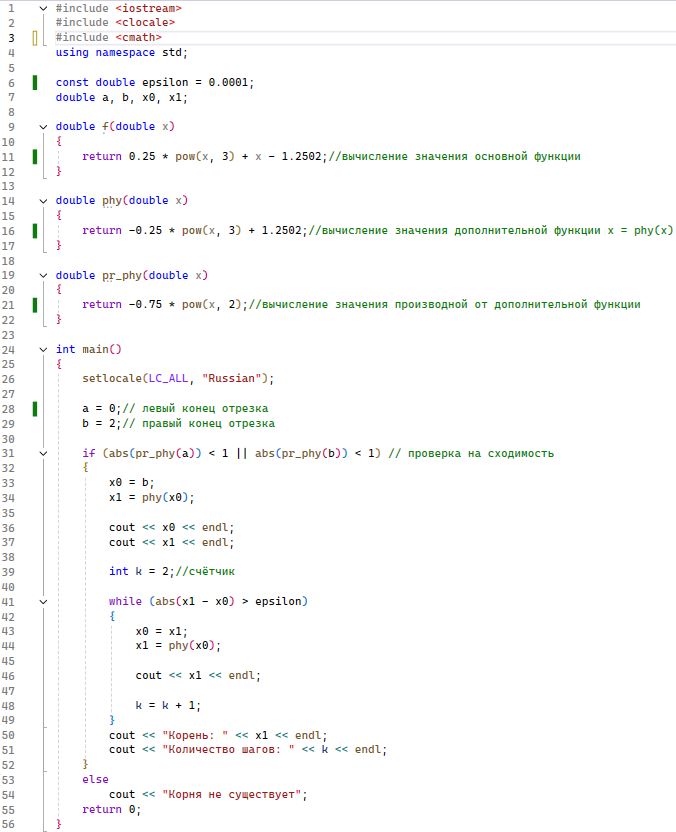
1. Установим начальное приближение x0 (x0 [0; 2]) правую границу интервала 2. Следующее значение x1 = φ(2)
2. Следующие приближение x вычисляется по формуле xi = φ(xi-1)
3. Это приближение будет повторятся до тех пор, пока |xn+1-xn| > Epsilon
4. Если |xn+1-xn| <= Epsilon, тогда завершаем цикл и выводим последнее x, так как это наше искомое значение.

Блок-схема:

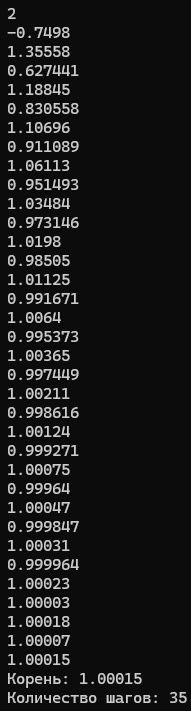




Код в С++:



Отработка программы:



Ссылка на github: