

# **Решение уравнений методом половинного деления**

Численный поиск корня: отрезок с разными знаками сужается  
пополам до нужной точности.

# Метод половинного деления на уроках информатики

На информатике в 10 классе этот метод помогает увидеть, как алгоритм на практике решает уравнения, когда точная формула недоступна: через циклы, условия и аккуратную работу с вещественными числами и погрешностями. Он подходит, например, для  $x^3 - x - 2 = 0$  и  $\cos(x) = x$ .

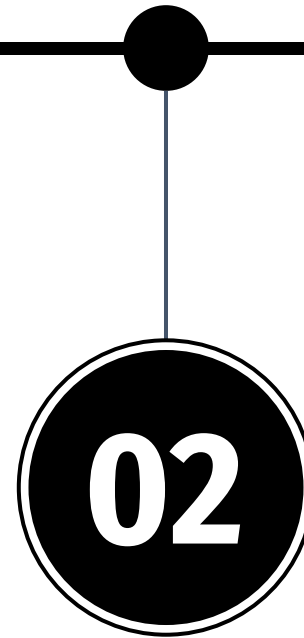


# Смысл численного поиска корня



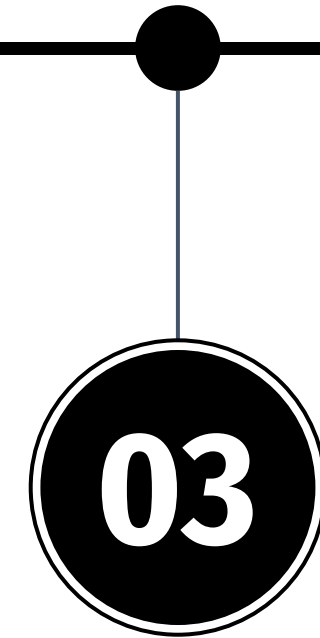
## Корень как равенство

Корень уравнения — это значение переменной, при котором левая и правая части совпадают. Для функции  $f(x)$  корнем называют  $x$ , когда  $f(x)=0$ .



## Поиск на отрезке

Метод половинного деления работает на отрезке  $[a,b]$ , где функция меняет знак:  $f(a)$  и  $f(b)$  имеют разные знаки. Тогда внутри промежутка гарантируется наличие хотя бы одного корня.



## Сужение области

Вместо «мгновенного» вычисления берут середину  $c=(a+b)/2$  и проверяют  $f(c)$ . Затем отрезок сокращают вдвое, повторяя шаг, пока область неопределенности не станет достаточно малой.

# Геометрический смысл метода

## Пересечение графика с осью $Ox$

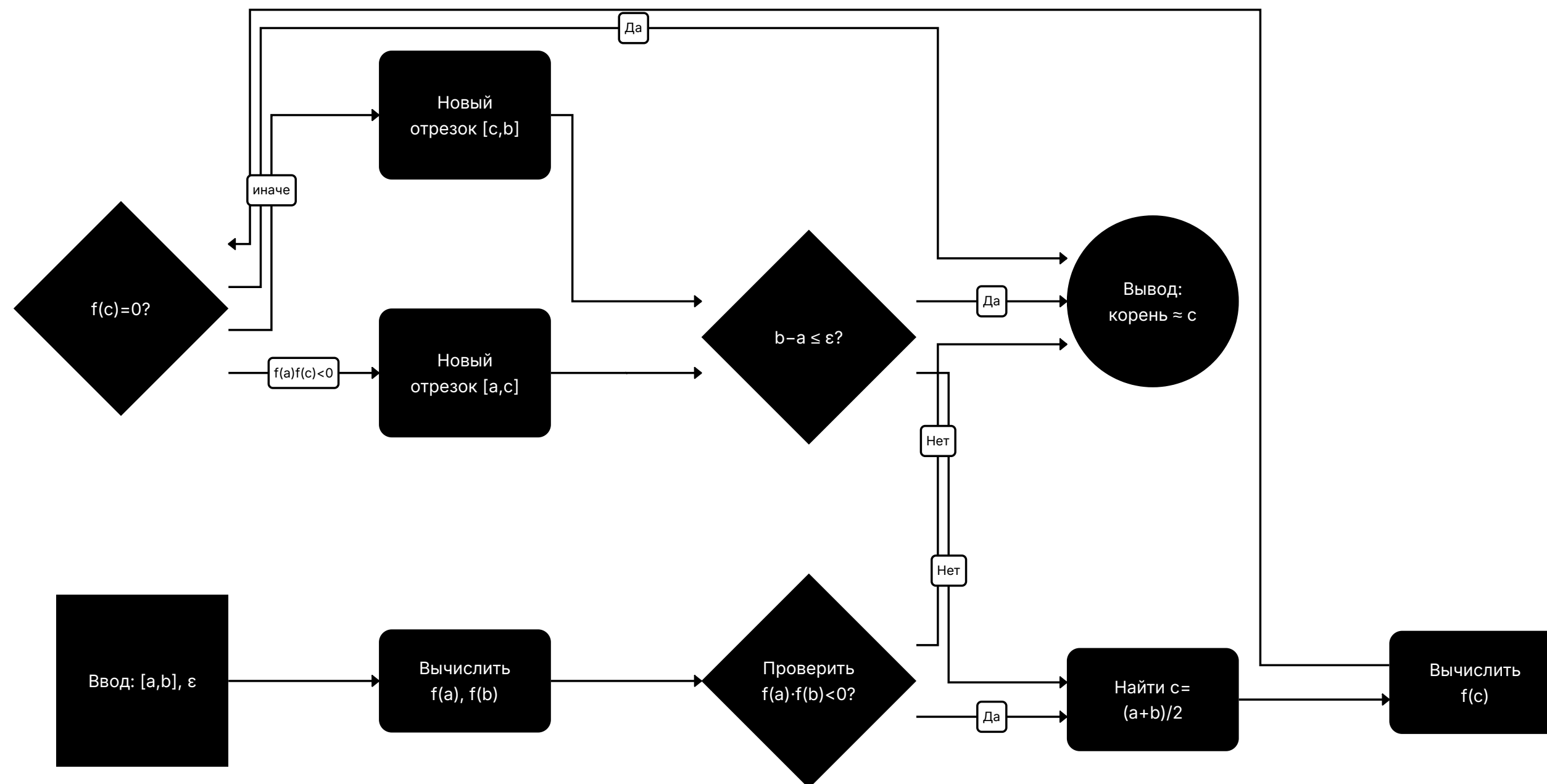
Корень уравнения соответствует точке, где график функции пересекает ось  $Ox$ : там значение функции равно нулю. Если на концах отрезка  $f(a)$  и  $f(b)$  имеют разные знаки, пересечение обязательно происходит внутри промежутка.

## Выбор половины отрезка

На каждом шаге находят середину  $c=(a+b)/2$  и вычисляют  $f(c)$ . Далее выбирают ту половину отрезка, на которой снова сохраняется смена знака, то есть корень остаётся внутри. Промежуток последовательно сужается.

# Алгоритм половинного деления

Дано: отрезок  $[a,b]$  и требуемая точность  $\varepsilon$ . На каждом шаге выбирается половина, где функция меняет знак.



# Условия применения метода

Метод половинного деления корректен только для непрерывной функции на выбранном отрезке. Это важно, потому что именно непрерывность гарантирует наличие пересечения с осью  $Ox$ , а значит и корня.

На концах отрезка требуется смена знака:  $f(a) \cdot f(b) < 0$ . Тогда внутри промежутка лежит как минимум один корень. Если этого условия нет, метод не даёт обоснованного результата.

Если на отрезке функция имеет разрыв, возможна ситуация, когда знак меняется «в обход» корня. Тогда вычисления могут привести к неверному выводу о месте нахождения решения.

Метод работает стабильно, когда корень не является кратным и значение функции действительно меняет знак. Плюс нужно учитывать округления при работе с вещественными числами и задавать разумную точность.

# Сравнение методов поиска корня

Сравните требования и поведение трёх подходов: половинного деления, Ньютона и секущих. Таблица помогает выбрать метод под задачу.

Полезно начинать с половинного деления, когда нужна предсказуемость и нет производной. Для ускорения выбирают Ньютон или секущие, но они чувствительнее к начальным данным.

| Метод              | Производная | Сходимость                | Скорость/<br>удобство            |
|--------------------|-------------|---------------------------|----------------------------------|
| Половинное деление | Не нужна    | Почти всегда при условиях | Надёжно, но медленнее            |
| Ньютон             | Нужна       | Требует удачного старта   | Обычно быстрее                   |
| Секущие            | Не нужна    | Промежуточная             | Чаще быстрее половинного деления |

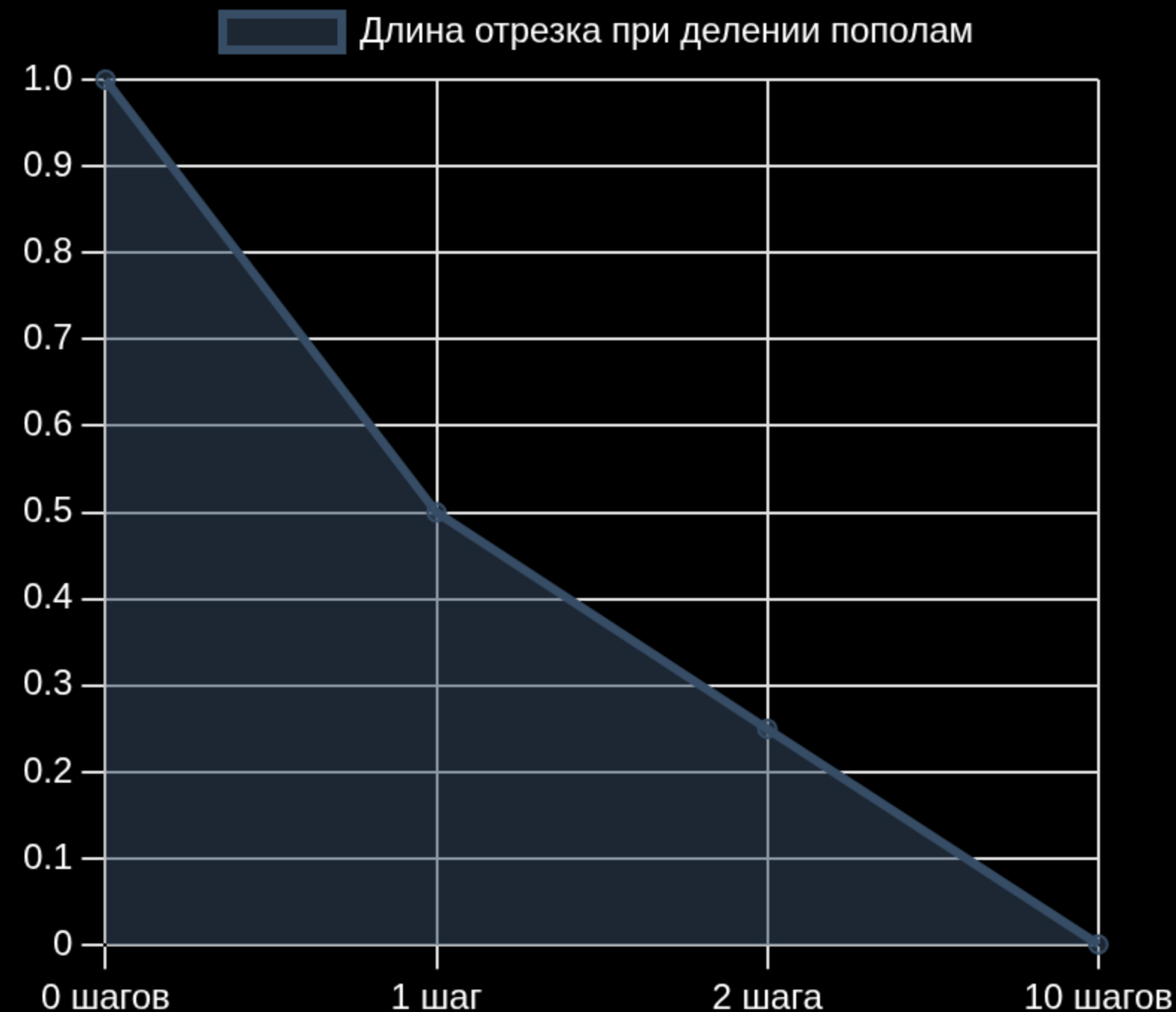
Сводное сравнение по характеристикам методов

# Как уменьшается длина отрезка при делении пополам

Каждый шаг уменьшает неопределённость вдвое: вы гарантированно сохраняете корень, но сокращаете область поиска в два раза. Поэтому растёт ожидаемая точность.

Быстрое геометрическое убывание делает метод удобным, чтобы сначала «локализовать» корень, а затем довести результат до нужной точности повторениями.

Теоретическая зависимость длины  $(b-a)/2^k$



# Оценка точности и погрешности

После  $n$  итераций корень гарантированно лежит внутри отрезка длиной  $(b-a)/2^n$ . Если приближением берут середину, абсолютная погрешность не превосходит половины этой длины.

Чтобы получить точность  $\varepsilon$ , достаточно выполнить условие  $n \geq \log_2((b-a)/\varepsilon)$ . Эта формула позволяет заранее посчитать, сколько шагов потребуется до требуемого уровня точности.

Например, при начальном отрезке длиной 1 и  $\varepsilon=0,001$  нужно не менее 10 итераций. Это соответствует тому, что  $2^{10} \approx 1024$ , а  $(1/2^{10})$  меньше 0,001.

# Пример выбора начального отрезка

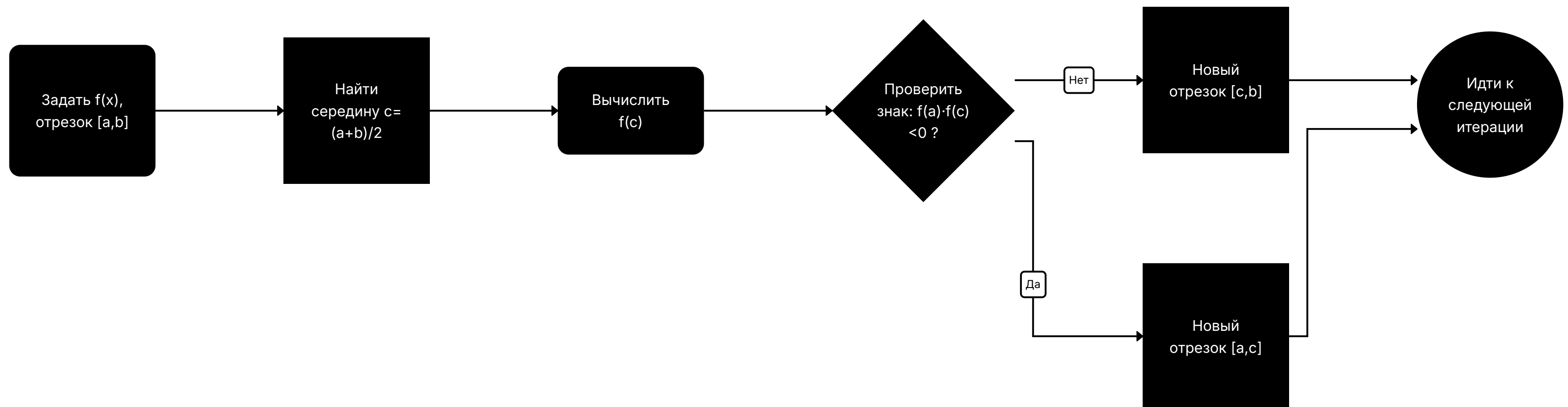
Для уравнения  $x^3 - x - 2 = 0$  удобно взять отрезок  $[1; 2]$ . Функция  $f(x) = x^3 - x - 2$  даёт  $f(1) = -2$  и  $f(2) = 4$ , то есть знаки на концах различны.

Разные знаки означает, что внутри промежутка существует как минимум один корень: условие метода выполнено. Значит, можно уверенно продолжать вычисления, сохраняя корень в выбранной области.

Такой пример часто используют в учебных заданиях, потому что числа простые, а ход алгоритма наглядный: отрезок будет последовательно сужаться к решению до нужной точности.

# Один шаг: делим отрезок и выбираем половину

Алгоритм для  $f(x)=x^3-x-2$  на старте  $[1;2]$  с решением внутри по смене знака



# Последовательность приближений к корню

01

После деления отрезка пополам получаются приближения: 1,5; 1,75; 1,625; 1,6875; 1,71875; 1,703125... На каждом шаге середина выбирается так, чтобы корень оставался внутри нового промежутка.

02

Для уравнения  $x^3 - x - 2 = 0$  эти значения постепенно «сгущаются» вокруг корня примерно 1,695. На графике это видно как движение точек середины к пересечению с осью  $Ox$ .

# Сходимость метода по числу шагов

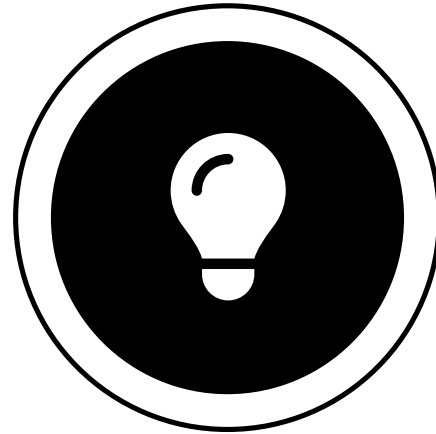
Верхняя граница ошибки при взятии середины примерно равна половине длины текущего отрезка:  $(b-a)/2^{n+1}$ . Поэтому снижение в 2 раза на каждом шаге видно даже в сравнении точек.

Диаграмма подтверждает предсказуемость: за 10 итераций неопределённость сокращается примерно в тысячу раз. Метод не самый быстрый, но даёт контролируемый рост точности.

Расчёт по формуле длины отрезка  $(b-a)/2^n$  при исходных  $a=1$ ,  $b=2$ ; без внешних данных, дата: 2026-05-14



# Преимущества метода



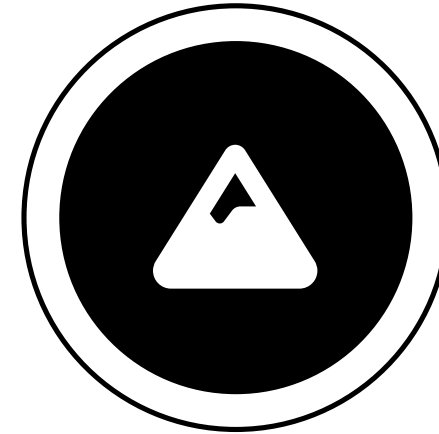
## Понятный алгоритм

Метод легко объяснить и записать как последовательность шагов: вычислить  $f$  в концах, взять середину, выбрать половину по знаку. Для школьного уровня это особенно наглядно.



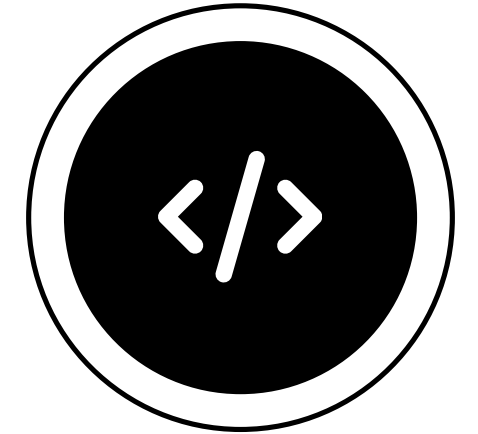
## Надёжность при корректных условиях

Если функция непрерывна на отрезке и на концах есть смена знака, корень гарантированно остаётся внутри выбранного промежутка. Благодаря этому результат получается устойчивым к «ошибкам выбора».



## Не нужна производная

В отличие от метода Ньютона, здесь не требуется производная и сложные преобразования. Это упрощает вычисления и делает метод удобным, когда функция задана формулой неудобно или таблично.



## Хорошо подходит для программирования

Метод естественно реализуется в цикле с условиями и переменными вещественного типа. Его можно писать на Python, Pascal или даже в электронных таблицах для быстрой отработки алгоритмического мышления.

# Ограничения и слабые стороны

## Условие знака

Нужна смена знака на концах отрезка:  $f(a) \cdot f(b) < 0$ . Если корень не отделён таким образом, метод может не сработать или дать бессмысленную последовательность.

## Не всегда найдёт кратный корень

Если корень кратный, функция может не менять знак. Тогда метод половинного деления не гарантирует обнаружение, даже при наличии корня внутри выбранного интервала.

## Численные округления

При вычислениях на компьютере накапливаются ошибки округления. Поэтому важно разумно задавать точность и не требовать излишне большое число знаков после запятой.

# Таблица итераций для уравнения $x^3 - x - 2 = 0$

Ниже показано, как меняются границы отрезка при каждом делении пополам и как знак  $f(c)$  определяет, какую половину оставить.

Каждая строка демонстрирует сохранение корня внутри интервала и постепенное уменьшение ширины неопределённости. При продолжении итераций границы сужаются к значению около 1,695.

| k | a   | b     | c      | f(c)    | новый отрезок  |
|---|-----|-------|--------|---------|----------------|
| 0 | 1   | 2     | 1,5    | -1,125  | [1,5;2]        |
| 1 | 1,5 | 2     | 1,75   | 1,6094  | [1,5;1,75]     |
| 2 | 1,5 | 1,75  | 1,625  | 0,3457  | [1,5;1,625]    |
| 3 | 1,5 | 1,625 | 1,5625 | -0,3533 | [1,5625;1,625] |

Собственные вычисления по формуле  $c = (a+b)/2$  и выбору половины по знаку  $f(a) \cdot f(c) < 0$ ; дата: 2026-05-14

# Связь между числом шагов и точностью

Если использовать середину отрезка как приближение, то гарантированная абсолютная погрешность не превосходит половины длины текущего интервала. Поэтому она убывает геометрически по числу шагов.

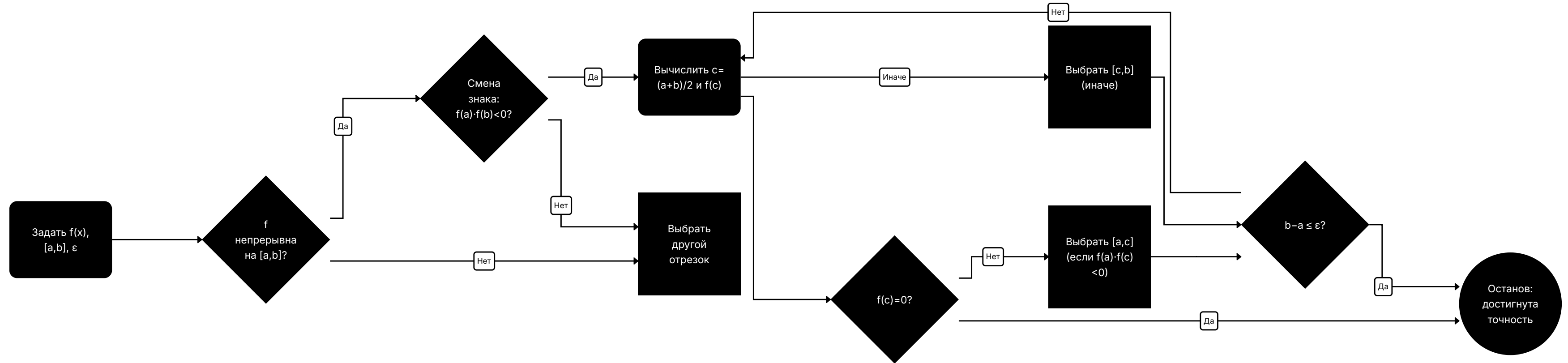
Данные показывают снижение ошибки в 2 раза при каждом добавленном шаге, что согласуется с теорией метода. После 10 итераций ошибка становится достаточно малой для большинства школьных задач.

Расчёт верхней границы:  $(b-a)/2^{n+1}$  при исходной длине  $b-a=1$  ( $a=1, b=2$ ); дата: 2026-05-14



# Проверка правильности вычислений

Порядок проверок и действий для корректного запуска метода половинного деления



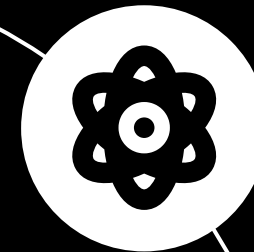
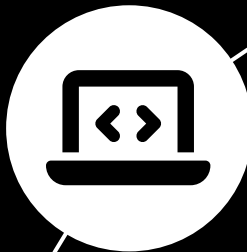
# Где применяется метод половинного деления

## Программирование

Метод используют как базовый пример вычислительного алгоритма: в коде легко организовать цикл, условия и работу с вещественными числами. Он подходит для задач, где формула решения неизвестна.

## Пороги и критерии

Во многих моделях важны точки, где функция меняет знак: например, границы режимов или момент «переключения». Метод помогает вычислить это значение с контролируемой точностью.

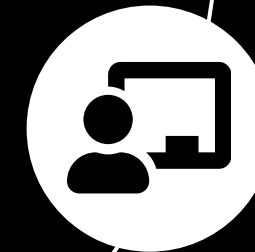


## Моделирование и физика

Когда уравнение описывает равновесие или переходы, корень определяют численно. Метод помогает находить решения по заданным параметрам, не требуя аналитических преобразований и сложной математики.

## Учебные цели в информатике

Метод половинного деления наглядно показывает связь между алгоритмом и точностью: каждое условное действие влияет на сужение интервала и контролирует ошибку. Это хороший фундамент для дальнейших численных методов.



## Табличные и формульные данные

Если функция задана формулой или таблицей, производную часто получить трудно. Деление пополам всё равно работает, пока можно вычислять  $f(x)$  и обеспечить смену знака на отрезке.

# **Метод половинного деления: простота, устойчивость и контролируемая точность**

Метод половинного деления — надёжный численный способ решения уравнений при непрерывности функции и смене знака на концах отрезка. Он ценится за простоту, устойчивость и предсказуемую точность, пусть и с умеренной скоростью сходимости.