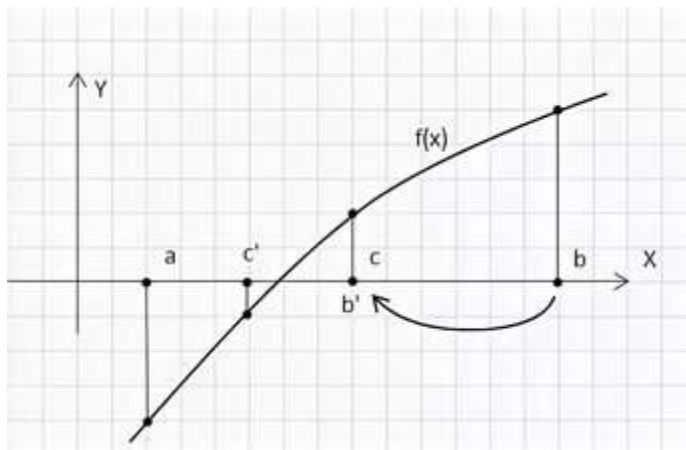


Метод бисекции (половинного деления)

Метод бисекции (половинного деления) — это численный метод для решения нелинейных уравнений вида $f(x) = 0$. Он основан на последовательном делении интервала пополам и выборе той половины, на концах которой функция сохраняет противоположные знаки.



Условия применения

Для применения метода необходимо, чтобы:

- функция $f(x)$ была непрерывной на искомом отрезке $[a; b]$;
- на концах отрезка функция принимала значения разных знаков, то есть выполнялось условие $f(a) \cdot f(b) < 0$.

Это гарантирует существование по крайней мере одного корня на интервале $[a; b]$ при условии непрерывности функции.

Алгоритм метода

1. Выбрать начальный интервал $[a; b]$, такой что $f(a) \cdot f(b) < 0$.
2. Вычислить середину интервала: $c = (a + b)/2$.
3. Если $f(c) = 0$ или длина интервала $|b - a|$ меньше заданной точности ε , то c — приближённое значение корня.
4. Если $f(a) \cdot f(c) < 0$, то заменить b на c .

5. Если $f(c) \cdot f(b) < 0$, то заменить a на c .
6. Повторять шаги 2–5 до достижения заданной точности.

Итерационный процесс продолжается до тех пор, пока длина отрезка не станет меньше заданной величины ε .

Преимущества и недостатки

Преимущества:

- простота реализации;
- гарантированная сходимость при выполнении условия $f(a) \cdot f(b) < 0$;
- не требует вычисления производной функции.

Недостатки:

- относительно медленная сходимость по сравнению с другими методами (например, методом Ньютона);
- необходимость знать интервал, содержащий корень

Программная реализация

Для программной реализации метода обычно включают следующие этапы:

- определение функции $f(x)$, для которой необходимо найти корень;
- задание начального интервала $[a; b]$ и точности ε ;
- проверка условия $f(a) \cdot f(b) < 0$; если условие не выполняется, необходимо изменить интервал или использовать другой метод;
- итерационный процесс: вычисление середины интервала, проверка знака произведения $f(a) \cdot f(c)$, обновление интервала в зависимости от знака произведения;

- проверка условия остановки: $|b - a| < \varepsilon$ или достигнуто максимальное количество итераций;
- возврат значения s в качестве приближённого значения корня.

Пример реализации на языке Python можно найти в источниках, например, в [источнике 7].

Метод бисекции легко реализуется на компьютере, но при увеличении точности значительно возрастает объём вычислений.