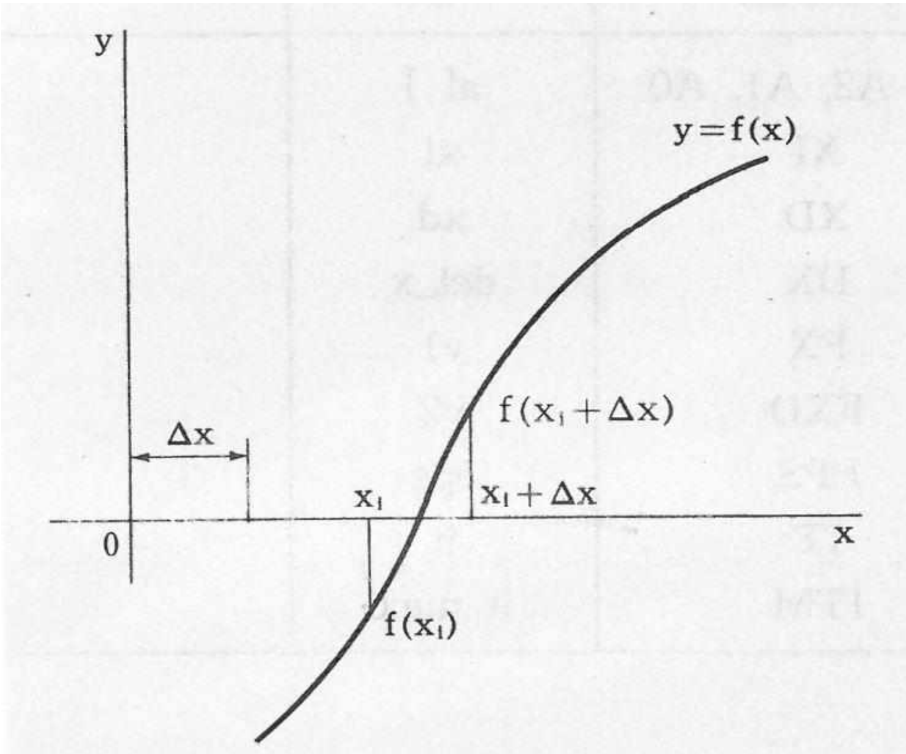
1. Bracketing Method(구간법)

: 함수가 근의 근처에서 부호가 변함에 기초한 방법으로 두 개의 초기값이 필요하며, 체계적으로 구간의 폭을 줄여 나가며 근에 접근하는 방법

▶ Incremental Search Method(증가탐색법, 증분검색법)

: 초기값에서 출발하여 조금씩 값을 증가하기를 반복하여 근을 구한다.



[Note]

1. 구간법의 초기 구간을 찾기 위해 사용된다.
2. Δx의 크기가 너무 작으면 탐색 소요 시간이 길어진다.
3. Δx의 크기가 너무 크면 근접한 해를 찾기가 어렵다.
4. 초기값을 잘못 설정하면 해를 구할 수 없다.

▷ 해를 찾을 수 없는 경우

1. Δx가 해의 간격보다 크면 두 개의 인접한 해를 놓칠 수 있다.
2. 중근인 경우
3. $f(x)$에 특이성이 있는 경우

[Algorithm]

(1) 초기근사해 x_i를 정한다.

(2) 초기증분 Δx를 정한다.

(3) $f(x_i)$ 및 $f(x_i+\Delta x)$를 계산한다.

(4) $f(x_i)*f(x_i+\Delta x)$의 부호를 계산한다.

(5) i) $f(x_i)*f(x_i+\Delta x)>0$ 이면

$x_i \leftarrow x_i+\Delta x$로 놓고 순서 (3)으로 간다.

ii) $f(x_i)*f(x_i+\Delta x)<0$ 이면

if $\Delta x \leq \varepsilon$이면, $x_i+\Delta x$를 근으로 하고 계산을 끝낸다.

else $\Delta x \leftarrow \Delta x * 0.1$로 하고, 순서 (3)으로 간다.

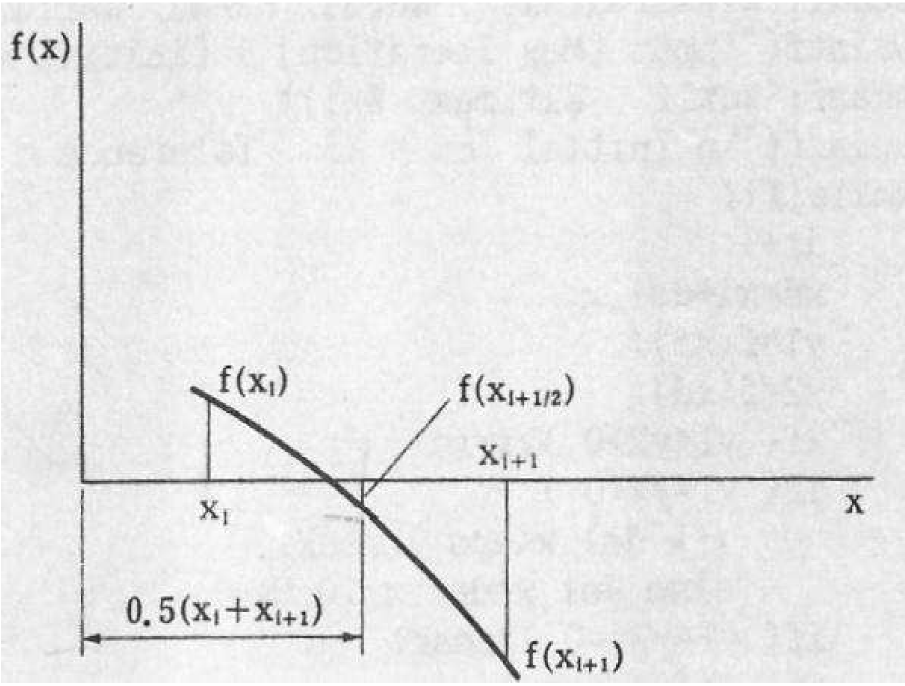
iii) $f(x_i)*f(x_i+\Delta_x)=0$ 이면

$x_i+\Delta x$를 근으로 하고 계산을 끝낸다.

[Example] $f(x)=x^3-x-1=0$, $x_0=1$, $\Delta x=0.1$, 4회 반복 후 x_i와 Δx는?

▶ Bisection Method(이분법)

: 근이 존재하는 구간을 반으로 나누고 나누어진 두 구간 중 근이 존재하는 구간을 다시 나누는 과정을 반복하여 근을 구한다. 이때, 근이 존재하는 구간의 가운데 위치한 값을 근이라 가정한다.



[Note]

1. 범위가 충분히 작아질 때까지 범위를 계속 정확히 절반으로 나눈다.
2. 발산의 위험이 없어 믿을 수 있는 확실한 방법이다.
3. 수렴 속도가 느리고 비효율적이다.
4. 다중 근의 경우 근을 찾지 못할 수도 있다.

[Algorithm]

(1) 구간 x_i, x_{i+1}을 입력한다.

(2) $f(x_i)$ 및 $f(x_{i+1})$을 계산한다.

(3) $x_{i+1/2} = (x_i + x_{i+1})/2.0$ 을 계산한다.

(4) $f(x_{i+1/2})$을 계산한다.

(5) $f(x_i) * f(x_{i+1/2})$의 부호를 계산한다.

i) >0 이면

$x_i \leftarrow x_{i+1/2}$로 하고 순서 (2)로 간다.

ii) <0 이면 $|x_{i+1} - x_i| < \varepsilon$를 확인하고

$|x_{i+1} - x_i| < \varepsilon$이면 $x_{i+1/2}$을 근으로 하고 계산을 끝낸다.

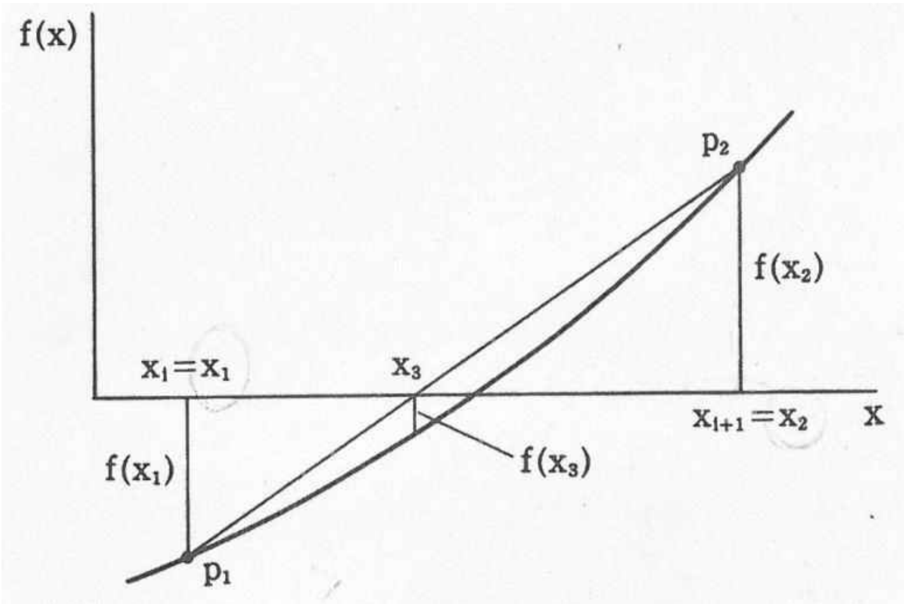
$|x_{i+1} - x_i| > \varepsilon$이면 $x_{i+1} \leftarrow x_{i+1/2}$로 하고 순서 (2)로 간다.

iii) $= 0$ 이면 $x_{i+1/2}$을 근으로 하고 계산을 끝낸다.

[Example] $f(x) = x^3 - x - 1 = 0$, 초기 구간이 [1,2]일 때, 2회 반복 후 근이 존재하는 구간은?

▶ The Method of False Position(가위치법)

: 근이 존재하는 구간의 두 끝점을 지나는 직선과 x축이 만나는 교점(근의 가위치) 을 구하는 과정을 반복하여 근을 구한다.



[Note]

1. 가위치는 삼각형의 닮음을 이용하여 구한다.

$$x_3 = \frac{x_1 f(x_2) - x_2 f(x_1)}{f(x_2) - f(x_1)}$$

2. 이분법보다 수렴 속도가 빠르다.
3. 곡률이 매우 큰 함수의 경우 이분법보다 느리다. (예. $f(x) = x^{10} - 1$)
4. 항상 근은 구간$(x_{1,}\ x_2)$ 내에 존재한다.
5. 한 방향으로만 접근하기 때문에 근에 무한히 접근하나 일치하기 어렵다.

[이분법과 비교]

★ 이분법 - 구간을 항상 반으로 나누며 함수값의 크기는 고려하지 않는다.

★ 가위치법 - 함수값이 0에 더 가까운 쪽의 x값이 근에 더 가까울 것을 고려.

[Algorithm]

(1) Determine $[x_1, x_2]$.

(2) Compute $f(x_1)$, $f(x_2)$.

(3) Compute x_3, $f(x_3)$.

(4) If $f(x_1)*f(x_3) > 0$,

if $|x_3 - x_1| < \epsilon$, x_3 : 근

else $x_1 \leftarrow x_3$, go to (2).

If $f(x_1)*f(x_3) < 0$,

if $|x_3 - x_2| < \epsilon$, x_3 : 근

else $x_2 \leftarrow x_3$, go to (2).

If $f(x_1)*f(x_3) = 0$, x_3 : 근

[Example] $f(x) = 2x^3 - 11x^2 + 2x + 15 = 0$, $x_1 = 0$, $x_2 = 4$일 때, 근은?

[Example] $f(x) = x^3 - x - 1 = 0$, 초기 구간 $[1,2]$