Metody numeryczne

Wykład 2

Michał Łaskawski

Wprowadzenie

Sposób postępowania

Numeryczne rozwiązywanie układów nieliniowych równań algebraicznych wykorzystują zależności rekurencyjne, które określają sposób obliczania kolejnych wyrażeń ciągu przybliżeń miejsc zerowych funkcji $f(x)$:

$$f(x) = 0 \tag{1}$$

Zakłada się, że równanie (1) posiada jedynie pierwiastki odosobnione.

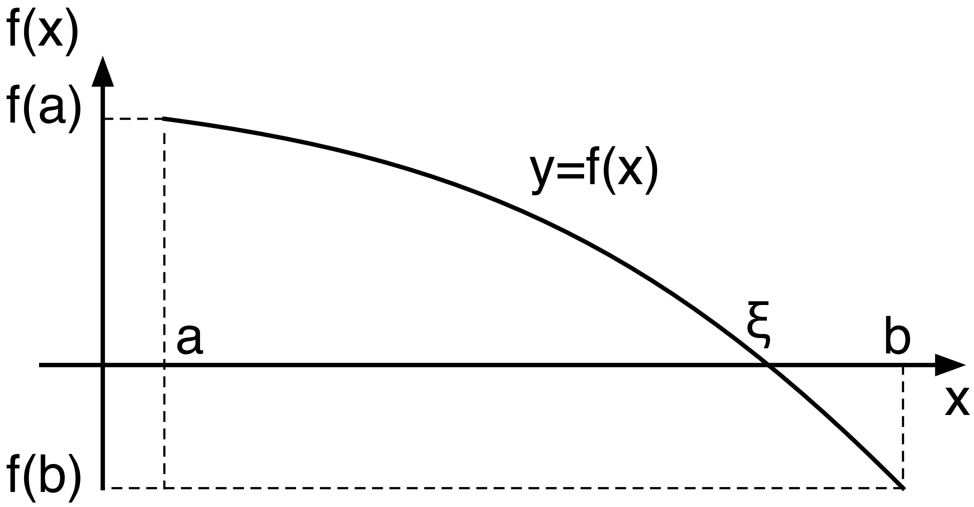
Pierwiastki odosobnione to takie pierwiastki, które zlokalizowane są wewnątrz przedziałów (a, b). Każdy taki przedział może zawierać tylko jeden pierwiastek.

Wprowadzenie

Lokalizacja pierwiastka

Twierdzenie Bolzano – Couchego:

Jeżeli funkcja $f(x)$ jest ciągła w przedziale domkniętym $[a, b]$ oraz $f(a)\,f(b) < 0$, to wewnątrz takiego przedziału znajduje się co najmniej jeden pierwiastek równania $f(x) = 0$. To znaczy istnieje co najmniej jedna liczba $\xi \in (a, b)$, dla której $f(\xi) = 0$.

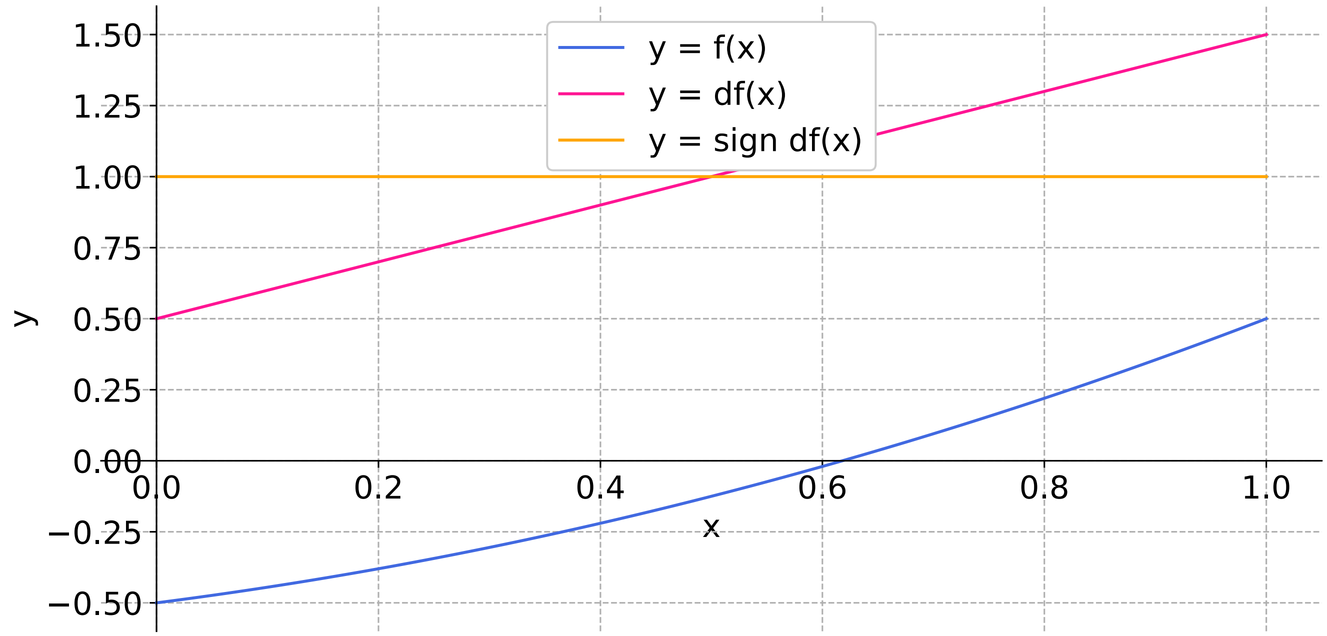


Wprowadzenie

Lokalizacja pierwiastka

Przedział izolacji pierwiastka:

Jeżeli w przedziale otwartym (a, b) istnieje pochodna $f^{(1)}(x)$ i nie zmienia ona znaku, to znaczy: $\textbf{sign}\, f^{(1)}(x) = const$ dla $x \in (a, b)$, to przedział (a, b) jest przedziałem izolacji pierwiastka.



Metoda połowienia przedziałów

Założenie

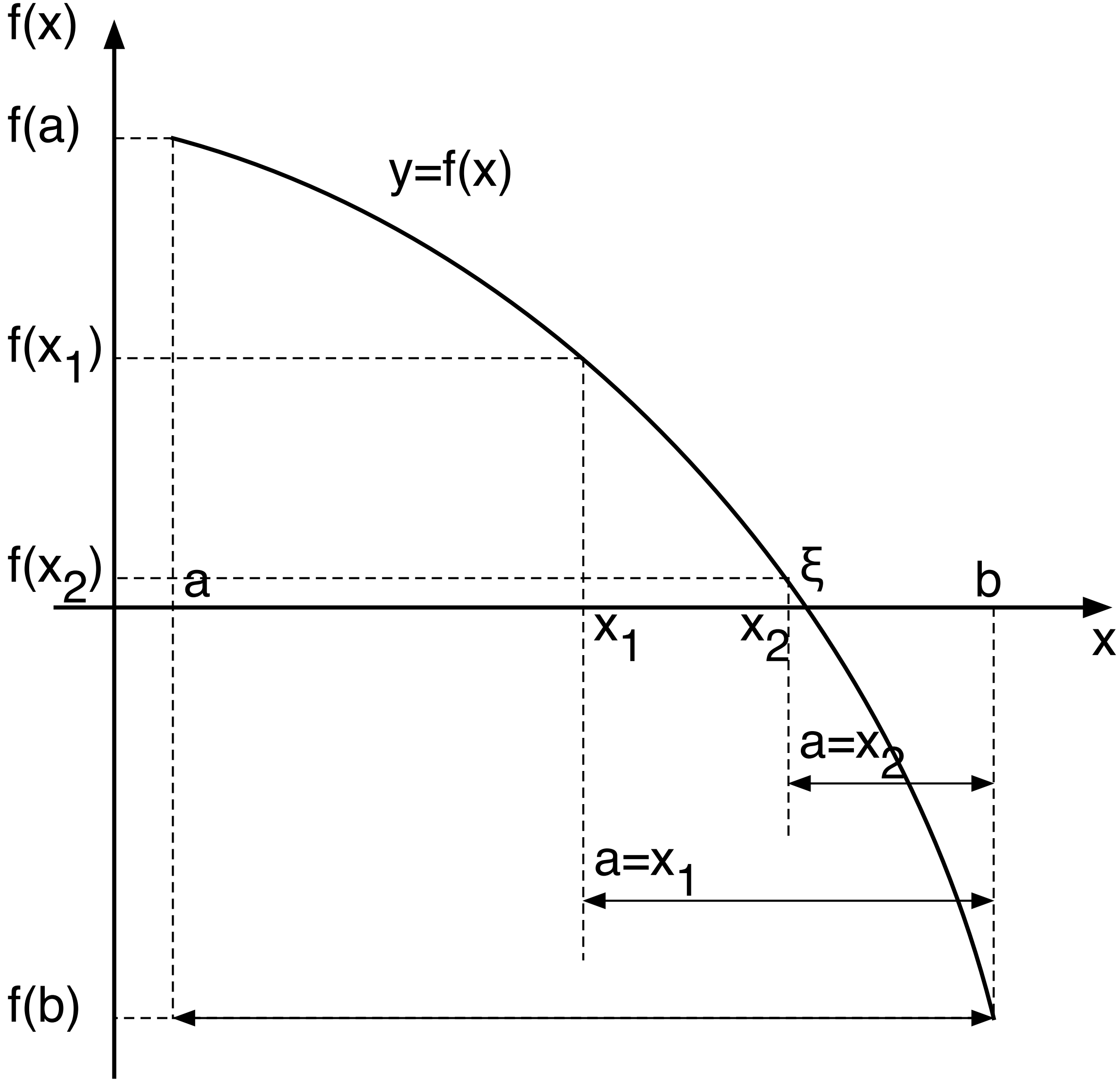
Dane jest równanie (1): $f(x) = 0$, przy czym funkcja $f(x)$ jest ciągła i monotoniczna w przedziale izolacji $[a, b]$ pierwiastka i zachodzi nierówność: $f(a)\,f(b) < 0$.

Algorytm

1. Podzielić przedział $[a, b]$ na połowę, środek przedziału: $x_1 = \frac{1}{2}(a + b)$.
2. Jeżeli $f(x_1) = 0$, to $\xi = x_1$ jest pierwiastkiem równania (1).
3. Jeżeli $f(x_1) \neq 0$, to odrzuć ten przedział $[a, x_1]$ lub $[x_1, b]$, który nie spełnia zależności $f(x_1)\,f(a) < 0$ lub $f(x_1)\,f(b) < 0$.
4. Oznacz granicę nowego przedziału jako $[a, b]$ i powróć do kroku 1.

Metoda połowienia przedziałów

Ilustracja działania metody



Metoda połowienia przedziałów

Przykładowa implementacja w języku Python

```
def f(x):
    "Definicja funkcji”
    return 0.5 * x ** 2 + 0.5 * x - 0.5

# Metoda bisekcji
iter = 100
delta = 0.00001
a = -2.25
b = 1.5
for k in range(1, iter):
    x = (a + b) / 2
    if abs(f(x)) < delta:
        break
    else:
        if f(x) * f(a) < 0:
            b = x
        else:
            a = x
```

Metoda siecznych

Funkcja klasy $C^{(n)}$

Jeżeli funkcja f ma w przedziale (a, b) n pochodnych i n -ta pochodna jest funkcją ciągłą w przedziale (a, b), to funkcję f nazywa się funkcję klasy $C^{(n)}(a, b)$.

Założenie

Niech $f(x)$ będzie funkcją klasy $C^{(2)}$ dla $x \in [a, b]$ oraz $f(a)\, f(b) < 0$.

W celu wyznaczenia przybliżenia rozwiązania równania (1): $f(x) = 0$, należy poprowadzić sieczną przechodząca przez dwa punkty: $A\big(a, f(a)\big)$ oraz $B\big(b, f(b)\big)$. Punkt przecięcia siecznej z osią OX jest przybliżeniem rozwiązania. Równanie siecznej:

$$\frac{x - a}{b - a} = \frac{y - f(a)}{f(b) - f(a)} \tag{2}$$

Metoda siecznych

Podstawiając do równania (2): $x = x_1$ oraz $y = 0$, otrzymuje się zależność określającą pierwsze przybliżenie rozwiązania:

$$x_1 = a - \frac{f(a)}{f(b) - f(a)}(b - a) \tag{3}$$

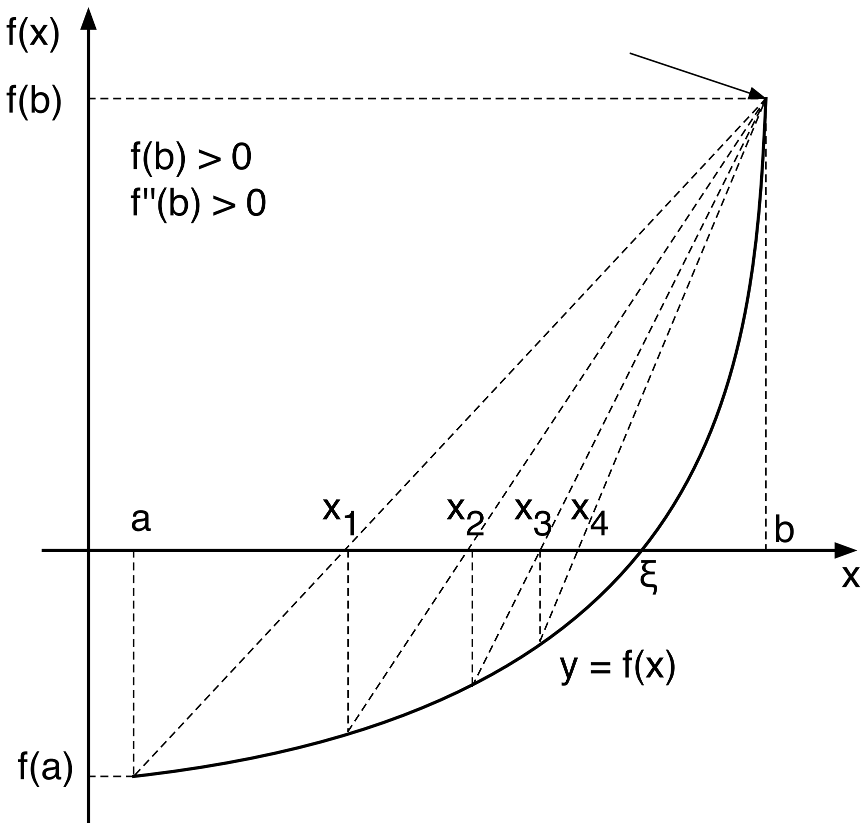
Postępując w ten sposób, dla kolejno zawężanych przedziałów $[x_n, b]$ dla $n = 1,2, \dots$ (lub $[a, x_n]$, w zależności od wybranego punktu stałego), otrzymuje się kolejne przybliżenia punktu stałego.

Uwaga:

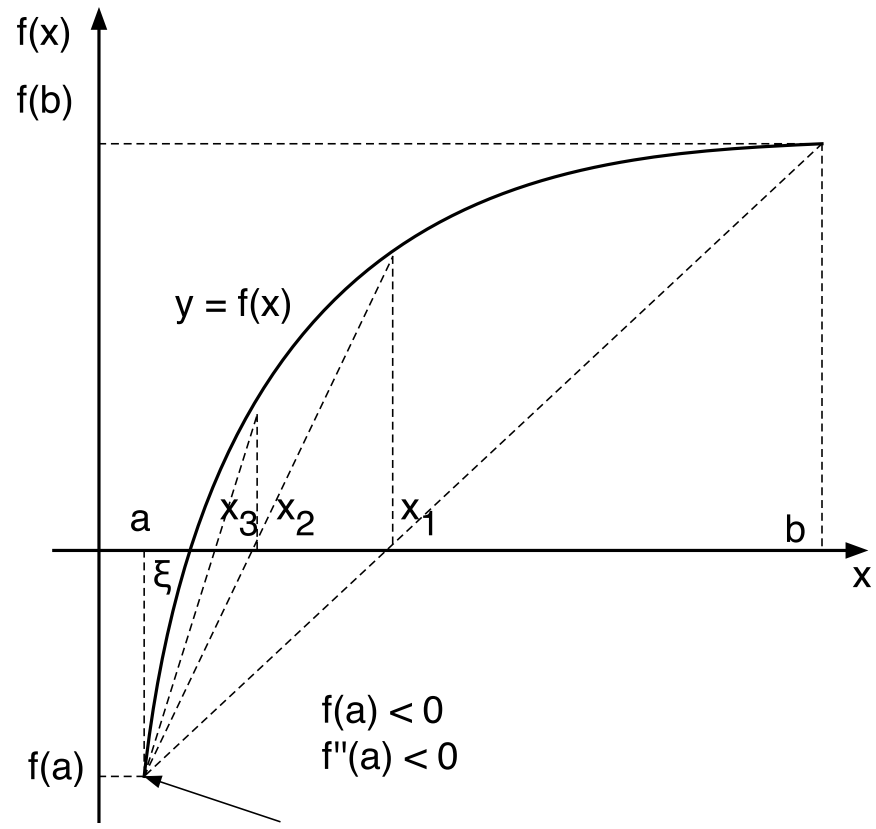
Jeżeli $f^{(2)}(x)$ nie zmienia znaku w przedziale $[a, b]$, to zbieżność metody zależy od wyboru jednego z punktów przedziału w którym funkcja $f(x)$ oraz jej druga pochodna $f^{(2)}(x)$ mają ten sam znak.

Metoda siecznych

Kolejne przybliżenia x_n leżą po tej samej stronie pierwiastka ξ, po której funkcja $f(x)$ oraz jej druga pochodna $f^{(2)}(x)$ mają różne znaki.



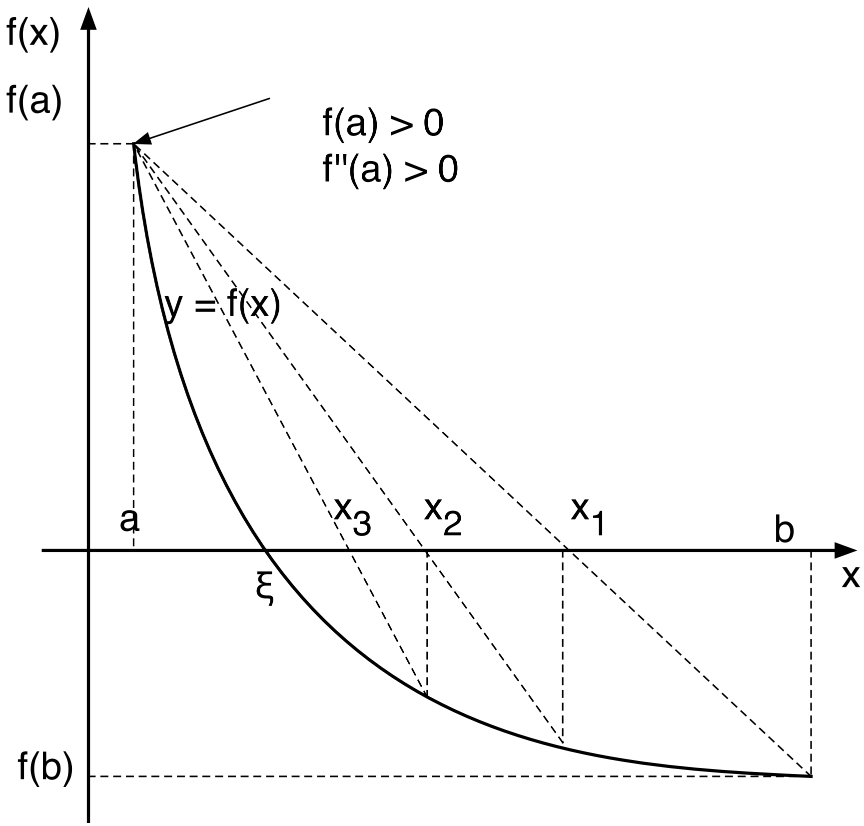
$f(a)\,f(x_n) > 0$



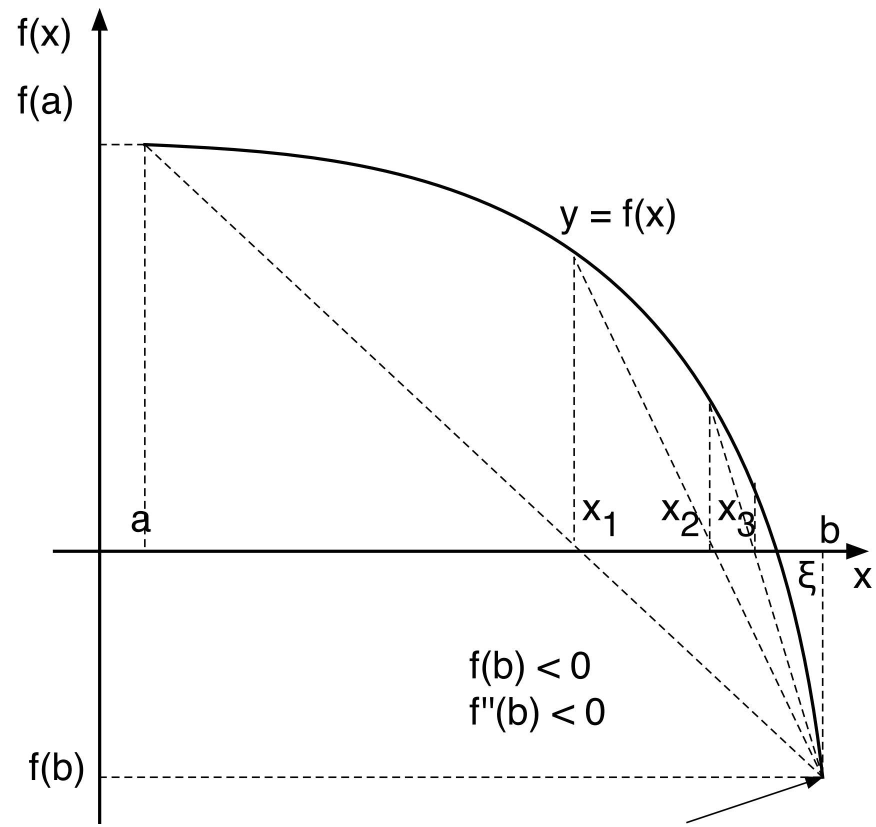
$f(a)\,f(x_n) < 0$

Metoda siecznych

Kolejne przybliżenia x_n leżą po tej samej stronie pierwiastka ξ, po której funkcja $f(x)$ oraz jej druga pochodna $f^{(2)}(x)$ mają różne znaki.



$$f(a)\, f(x_n) < 0$$



$$f(a)\, f(x_n) > 0$$

Metoda siecznych

Przykładowa implementacja w języku Python

```
def f(x):
    "Definicja funkcji”
    return 0.5 * x ** 2 + 0.5 * x - 0.5

# Metoda siecznych
iter = 100
delta = 0.00001
a = -2.25
b = 1.5
for k in range(1, iter):
    x = a - (f(a) / (f(b) - f(a))) * (b - a)
    if abs(f(x)) < delta:
        break
    else:
        if f(a) * f(x) > 0:
            a = x
        else:
            b = x
```

Metoda stycznych (Newtona)

Założenie:

Dane jest równanie (1): $f(x) = 0$, które posiada pierwiastek xi w przedziale $[a, b]$, w którym funkcje $f^{(1)}(x)$ oraz $f^{(2)}(x)$ są ciągłe i nie zmieniają znaku.

Metoda Newtona polega na przybliżeniu rozwiązania równania $f(x) = 0$ wyrazami ciągu miejsc zerowych stycznych do $f(x)$.

Równanie stycznej w punkcie: $B_n\left(x_n, f(x_n)\right)$, $n = 1,2, \ldots$

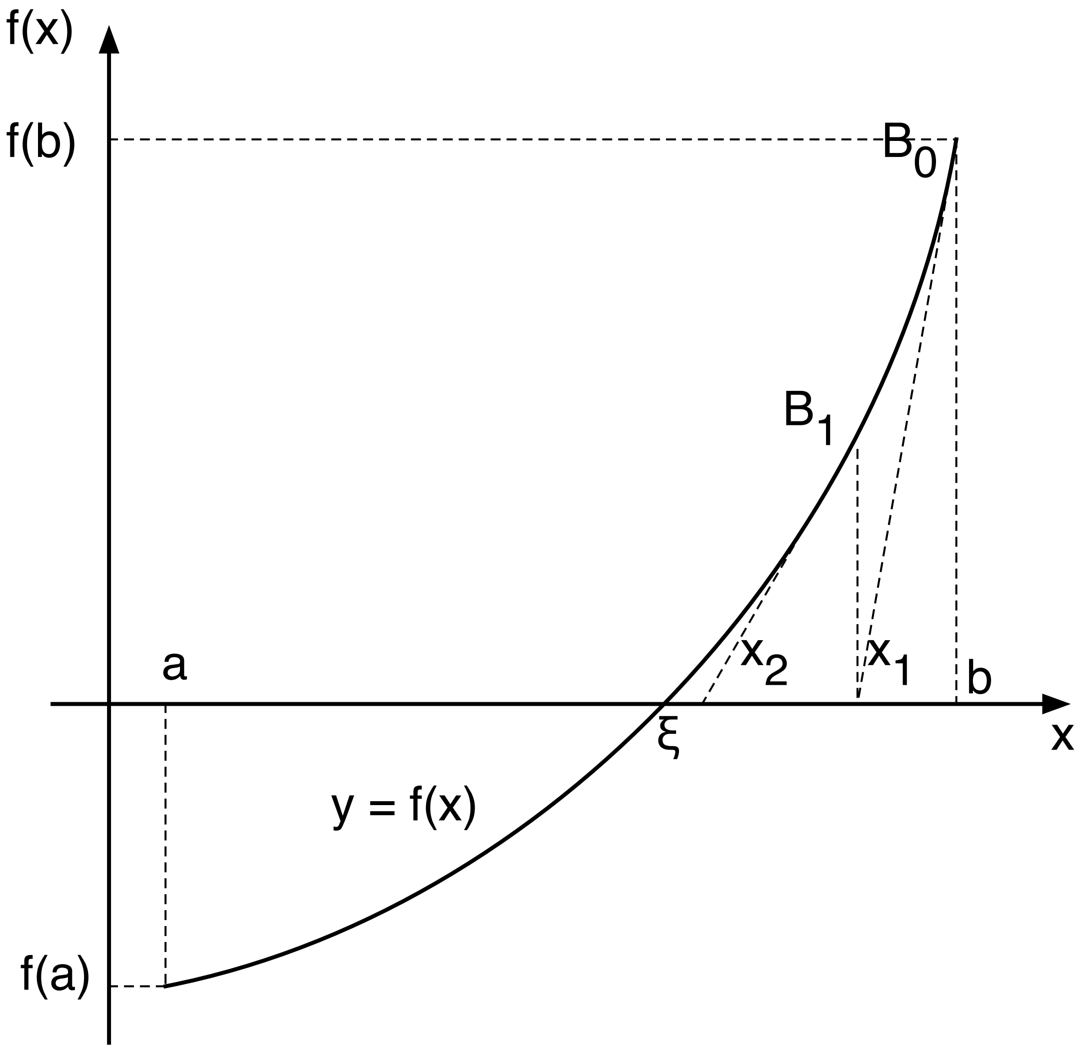
$$y - f(x_n) = f^{(1)}(x_n)(x - x_n) \tag{4}$$

Przyjmując, że $y = 0$ oraz $x = x_n + 1$, otrzymuje się

$$x_{n+1} = x_n - \frac{f(x_n)}{f^{(1)}(x_n)} \tag{5}$$

Metoda stycznych (Newtona)

Ilustracja działania metody



Metoda stycznych (Newtona)

Zbieżność

Zbieżność metody w dużym stopniu zależy od wyboru wartości początkowej $x_0 = a$ (lub $x_0 = b$).

Stosując metodę stycznych należy kierować się zasadą: jako punkt początkowy x_0 wybiera się koniec przedziału $[a, b]$ dla którego $f(x_0)f^{(2)}(x_0) > 0$.

Metoda Newtona jest szczególnie efektywna, gdy funkcja $f(x)$ jest stroma w otoczeniu punktu ξ.

Nie zaleca się stosować tej metody gdy krzywa $f(x)$ w pobliżu przecięcia osi OX jest prawie pozioma.

Metoda stycznych (Newtona)

Przykładowa implementacja w języku Python

```
def f(x):
    "Definicja funkcji”
    return 0.5 * x ** 2 + 0.5 * x - 0.5

def df(x, dx):
    "Definicja aproksymacji pierwszej pochodnej”
    return (f(x + 0.5*dx) - f(x-0.5*dx)) / dx

# Metoda stycznych
iter = 100
delta = 0.00001
dx = 0.001
x = -2.25
for k in range(1, iter):
    x = x - f(x) / df(x, dx)
    if abs(f(x)) < delta:
        break
```