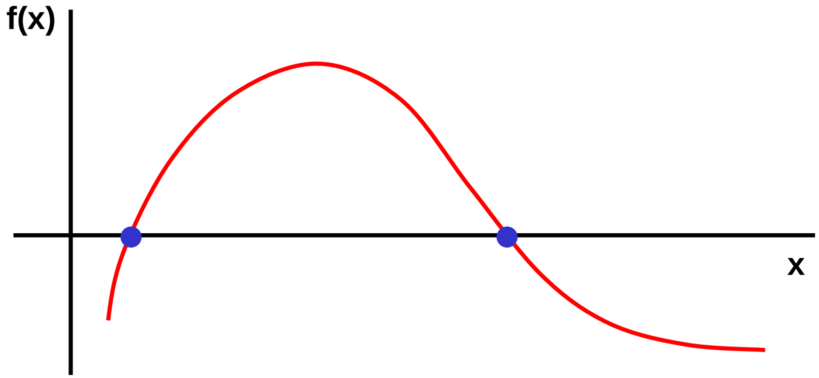
Rozwiązywanie równań nieliniowych (szukanie ich pierwiastków)



f(x) jest znana

$f(x_r)= 0$ szukamy $x_r = ?$

I. Metody izolacji pierwiastków (Bracketing Methods) (Pierwiastek jest wyizolowany w przedziale [a, b] jeżeli f(a) oraz f(b) mają przeciwne znaki (wyjątek dla funkcji z osobliwościami) Metoda jest zawsze zbieżna.)

a) Metoda równego podziału (metoda połowienia, metoda bisekcji, metoda połowienia przedziału) (Bisection Method)

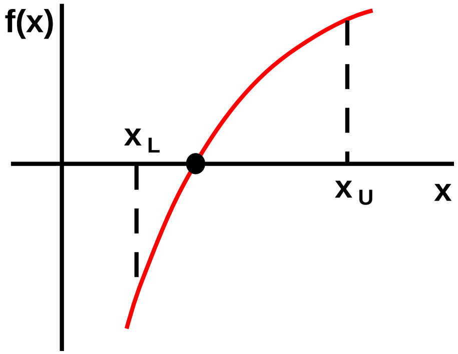
b) Regula falsi (False-Position Method)

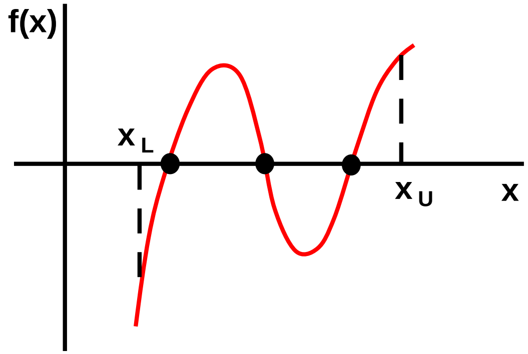
II. Metody otwarte (Open Methods) (Potrzebują jeden lub dwa wstępnie oszacowane punkty. Metoda może być rozbieżna)

a) Metoda Newtona-Raphsona (metoda stycznych) (Newton-Raphson Method) (Musimy znać pochodną funkcji.)

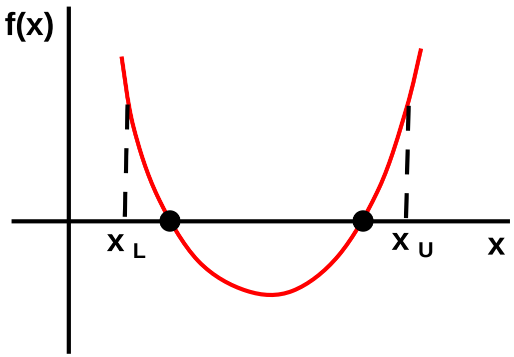
b) Metoda siecznych (metoda Eulera) (Secant Method)

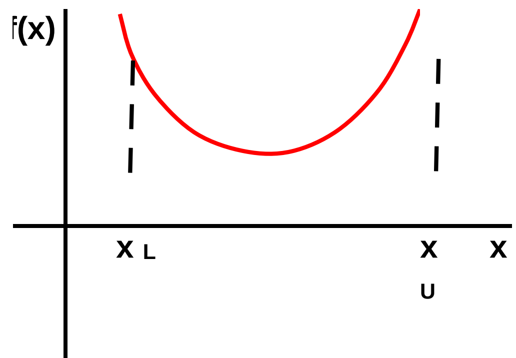
Ogólna idea bracketing



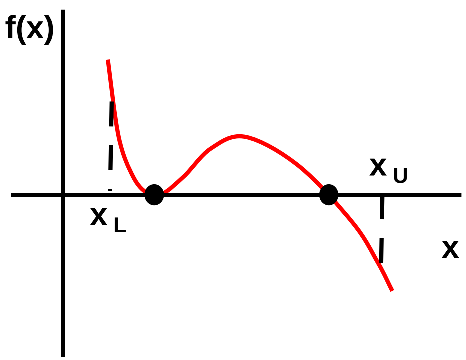


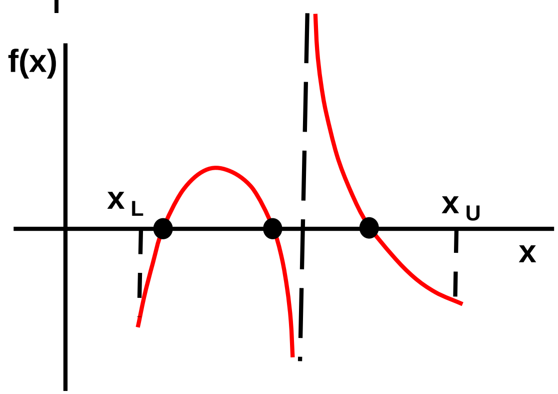
Reguła 1: Jeśli $f(x_L)*f(x_U) < 0$ mamy nieparzystą ilość pierwiastków





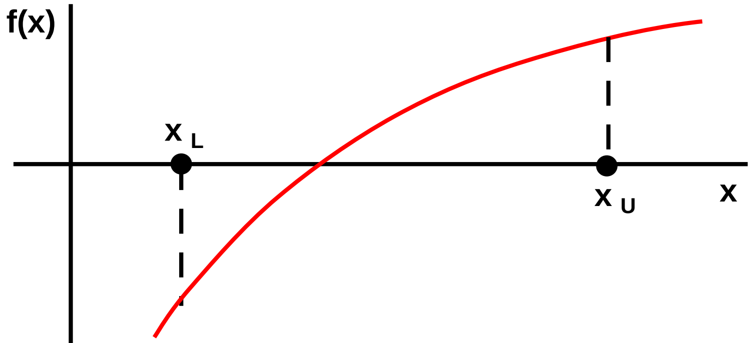
Reguła 2: Jeśli $f(x_L)*f(x_U) > 0$ mamy parzystą ilość pierwiastków **lub brak**



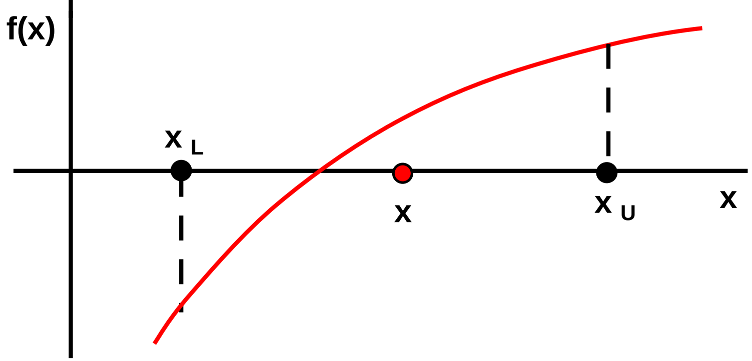


Naruszenia reguł: (a) pierwiastki wielokrotne, (b) nieciągłości funkcji (osobliwości)

Metoda połowienia



- Startujemy z dwóch punktów, x_{LOWER} and x_{UPPER} .
- Powinny zawierać pierwiastek, tzn. $f(x_L) * f(x_U) < 0$

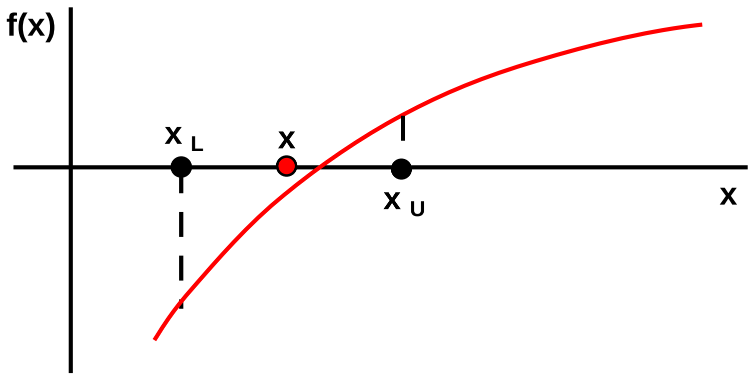


- Szacujemy jako punkt środkowy przedziału.

$$\boxed{x = (x_L + x_U)/2}$$

- Określamy przedział który zawiera pierwiastek, lewy

jeśli $f(x_L) * f(x) < 0$, w przeciwnym razie prawy



- Powtarzamy obliczenia dla nowego przedziału
- Zatrzymujemy po osiągnięciu określonej tolerancji

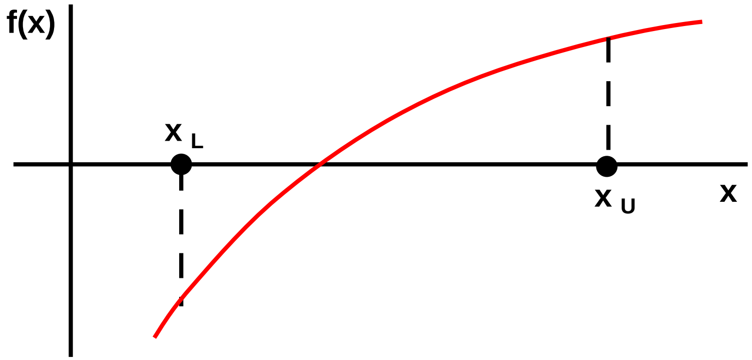
Przykład 1: Znajdź pierwiastek kwadratowy z 11.

$x^2 = 11$ -> $f(x) = x^2 - 11$ (dokładne rozwiązanie 3.31662479)

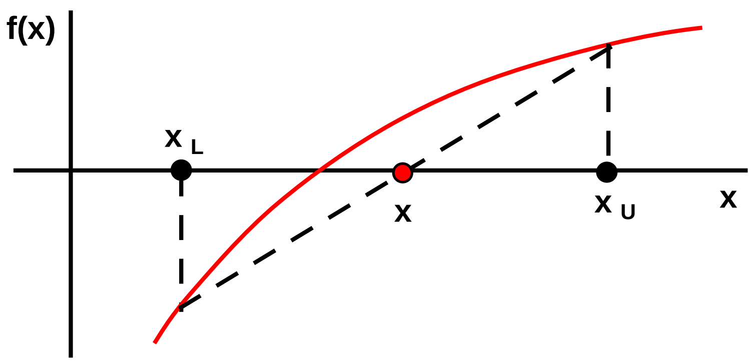
Wybieramy punkty początkowe: $3^2=9 < 11$, $4^2=16 > 11$ -> $x_L = 3$, $x_U = 4$

Iteracja.	x	f(x)	$\lvert e_t \rvert$ %
1	3.5	1.25	5.53
2	3.25	-0.4375	2.01
3	3.375	0.390625	1.76
4	3.3125	-0.02734375	0.12
5	3.34375	0.180664062	0.82
6	3.328125	0.076416015	0.35

Reguła falsi

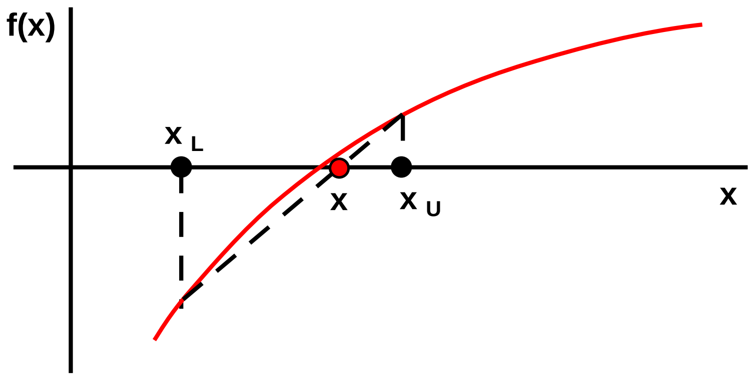


- Startujemy z dwóch punktów, x_{LOWER} and x_{UPPER} .
- Powinny zawierać pierwiastek, tzn. $f(x_L) * f(x_U) < 0$



- Szacujemy pierwiastek $$\boxed{x = x_U - \frac{f(x_U)(x_L - x_U)}{f(x_L) - f(x_U)}}$$

- Określamy przedział który zawiera pierwiastek, lewy jeśli $f(x_L) * f(x) < 0$, w przeciwnym razie prawy

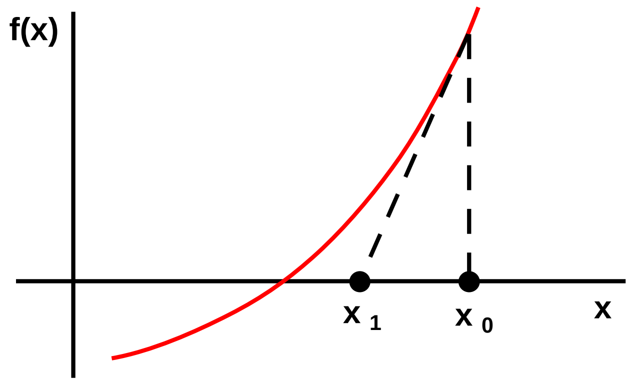


- Powtarzamy obliczenia dla nowego przedziału
- Zatrzymujemy po osiągnięciu określonej tolerancji

Przykład 2: Rozwiąż poprzedni przykład (Pierwiastek kwadratowy z 11).

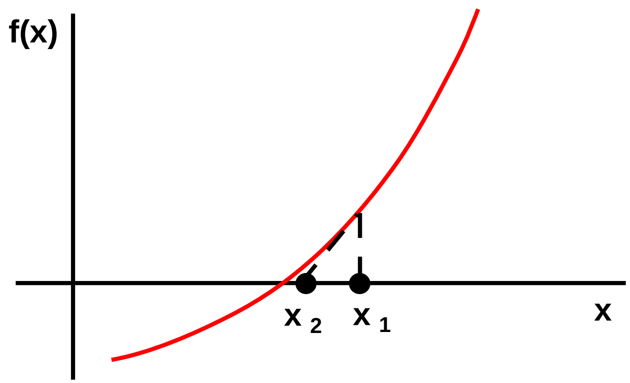
Iteracja	x	f(x)	$\|e_t\|$ %
1	3.28571429	-0.1040816	0.932
2	3.31372549	-0.0192234	0.087
3	3.31635389	-0.0017969	0.0082
4	3.31659949	-0.0001678	0.00076
5	3.31662243	-0.0000157	0.00007
6	3.31662457	-0.0000015	0.00001

Metody otwarte -Metoda Newton-Raphson



Startujemy z punktu, x_0 i obliczamy x_1

$$f'(x_0) = \frac{f(x_0) - 0}{x_1 - x_0} \quad \rightarrow \quad x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$



Kontynuujemy obliczenia do osiągnięcia określonej tolerancji lub ilości iteracji

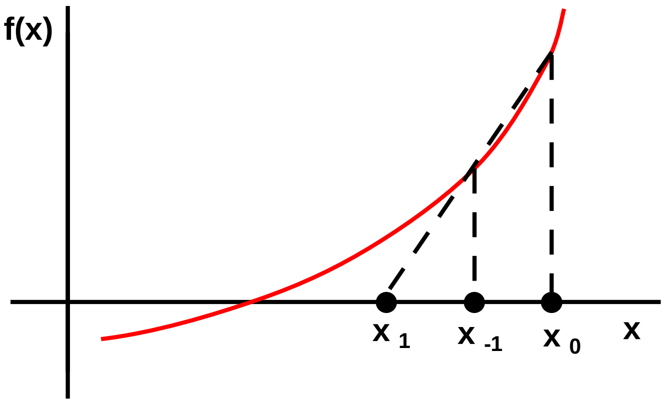
$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Przykład 3: Rozwiąż poprzedni przykład (Pierwiastek kwadratowy z 11).

Punkt początkowy $x_0 = 3$, $f(x) = x^2 - 11 = 0$, $f'(x) = 2x$ -> $x_{i+1} = x_i - (x^2 - 11)/2x$

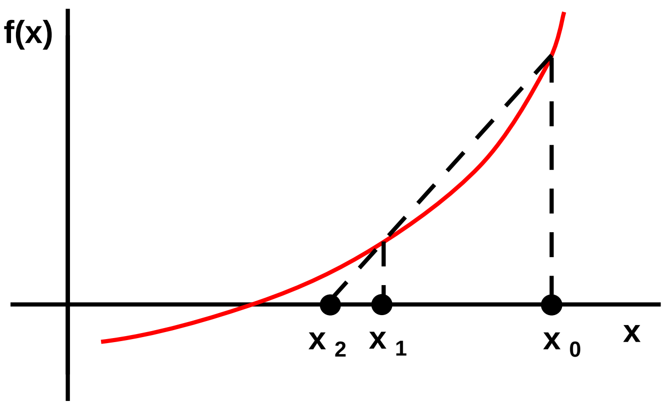
iteracja	x	f(x)	$\lvert e_t \rvert$ %
0	3	-2	9.55
1	3.33333333	0.1111111	0.50
2	3.31666667	0.0002778	0.00126
3	3.31662479	0.0000000	0.00000

Metoda siecznych



Startujemy z dwóch punktów, x_{-1} x_0

$$x_1 = x_0 - \frac{f(x_0)(x_{-1} - x_0)}{f(x_{-1}) - f(x_0)}$$



Kontynuujemy obliczenia do osiągnięcia określonej tolerancji lub ilości iteracji

$$x_{i+1} = x_i - \frac{f(x_i)(x_{i-1} - x_i)}{f(x_{i-1}) - f(x_i)}$$

Przykład 4: Rozwiąż poprzedni przykład (Pierwiastek kwadratowy z 11).

Punkty początkowe $x_{-1} = 2$, $x_0 = 3$, $f(x) = x^2-11 = 0$

iteracja	x	f(x)	$\lvert e_t \rvert$ %
-1	2	-7	-311
0	3	-2	-160
1	3.4	0.56	2.51
2	3.3125	-0.0273438	0.12
3	3.31657356	-0.0003398	0.0015
4	3.31662482	0.0000002	0.0000

Python – rozwiązanie równania nieliniowego metodą połowienia

```python
from scipy import optimize


def f(x):
    return (x**2 - 11)

xl=3
xu=4

root = optimize.bisect(f, xl, xu,full_output=True)

```

```
In [5]:

In [5]: runfile('C:/Users/u46/Desktop/root.py', wdir='C:/Users/u46/Desktop')

In [6]: root
Out[6]:
(3.3166247903554904,       converged: True
            flag: 'converged'
  function_calls: 41
      iterations: 39
            root: 3.3166247903554904)

In [7]:
```